2. Jahrgang – Nr 8/85 N 5263 E ISSN 0176-9367 Österreich 13 ÖS – Schweiz 1.50 sfr.

HCR Heim+ Personal Computer Report

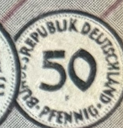
Die Microcomputer Zeitung

Für C 64 · VC 20 · Atari · ZX Spectrum
ZX 81 · TI 99/4 A · Colour Genie · Dragon
Schneider CPC 464 · Apple · Epson

1,50 DM

Textverarbeitung für





Superlisting für Schneider

Vergleich: Textverarbeitung

Abacus und Easel
Anwenderprogramme des QL

Kraftwerke
Computernetzteile

CPC 464 Supertext CPC 664
Biorhythmus für Apple II
Mathematik Teil 2 für Sinclair Spektrum
Sinclair QL: Die ersten Programme
Stadt in Not
Superspiel für CPC

8/85

Professionelle Software im Preis enthalten

Abacus, Archive, Easel, Quill

In der Ausgabe 6/85 von HCR - Heim Computer Report berichteten wir über den neuesten Rechner aus dem Hause Sinclair. Zu den besonderen Stärken des QL gehörten zweifellos die vier mitgelieferten Programme der englischen Softwarefirma PSION. Sie bestechen durch ihre Leistungsfähigkeit und leichte Handhabung und machen den Sinclair QL zu einem sofort einsetzbaren Werkzeug für kleine bis mittlere Unternehmen. Grund genug, diese Software genauer unter die Lupe zu nehmen.

von Klaus Weppler

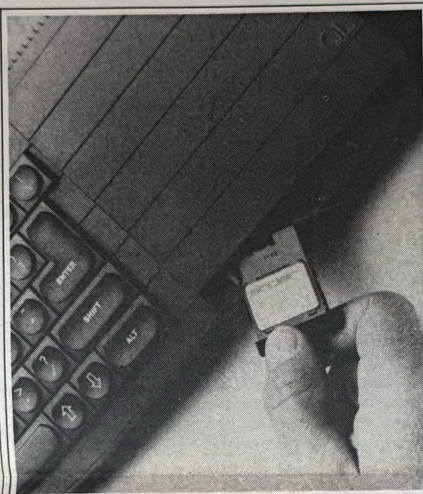
Die Summe von Sinclair QL und dem Softwarepaket von PSION bildet mit Sicherheit eine brauchbare Komplettlösung für den kleinen Geldbeutel. Ob man nun einen Brief schreiben, Adressen verwalten, Rechnungen verschicken, Bilanzen aufstellen oder diese graphisch darstellen will, es ist für jeden etwas dabei.

Alle vier Programme haben einige Gemeinsamkeiten, die der schnellen Einarbeitung sehr dienlich sind. Dazu gehört das Bildschirm - Layout. Das Bild ist aufgeteilt in drei Bereiche:

Kontroll - Bereich, der die Möglichkeiten anzeigt, die der Benutzer zu diesem Zeitpunkt hat, z. B. die Belegung der Funktionstasten, Cursorsteuerung und wählbare Kommandos. Dieser Bereich verschwindet auf Druck der Taste F2 und wird bei erneutem Druck wieder angezeigt. Der freigewordene Platz wird dem Arbeits - Bereich *in der Mitte zugeteilt.*

Das untere Fünftel beinhaltet die Eingabezeile und den Statusbereich zum Anzeigen des momentanen Arbeitsstandes, wie z. B. den restlichen Speicherplatz und Fehlermeldungen. Ebenfalls gleich bei allen vier Programmen ist der komfortable Zeileneditor und die Verwendung der Tasten F 1 - F 3 und ESC. Als besonders nützlich hat es sich erwiesen, daß man jederzeit durch Drücken der Taste F1 Benutzerinformationen auf den Bildschirm rufen kann. Zu diesem Zweck muß jedoch die Programmkassette ständig in Laufwerk 1 stecken.

Eine besondere Funktion hat die ESC-Taste. Mit ihr kann man jederzeit eine unvollständige Eingabe rückgängig machen oder in die nächst höhere Programmebene zurückkehren. Durch Drücken der Taste F3 erscheint im Kontrollbereich die Liste der wählbaren Kommandos.



Zur allgemeinen Verwendung der Programmkassetten sei noch Folgendes gesagt:

Um ein Zerstören der Originalkassetten zu verhindern, ist es zwingend nötig, eine Kopie zu erstellen und nur diese zu benutzen, da die eingebauten Microdrives des QL zu unzuverlässig arbeiten. Zum Kopieren der Programme befindet sich auf jeder Original - Kassette ein spezielles Backup - Programm, so daß dieser Vorgang keine Schwierigkeiten aufwirft.

Kommen wir nun zum Kalkulationsprogramm ABACUS. Mit einem ,,Rechenbrett'', wie es die alten Chinesen benutzten, hat dieses Programm nur zwei Dinge gemeinsam; es ist fast genauso leicht zu bedienen und man kann damit rechnen. Ansonsten - das können Sie mir glauben - ist ABACUS wesentlich leistungsfähiger.

Das ,,Arbeitsblatt'' des Programms besteht aus insgesamt 16 320 Zellen, die in einem Gitter von 64 Spalten und 255 Zeilen aufgeteilt sind. Davon wird jedoch immer nur ein Ausschnitt auf dem Bildschirm angezeigt. Die Breite einer Spalte beträgt nach dem Einschalten 10 Zeichen, sie ist jedoch mit dem GRID - Befehl zwischen 1 - 66 Zeichen frei einstellbar.

Mit den Cursortasten steuert man die einzelnen Zellen an und füllt diese mit einem Wert, einem Text oder einer Formel. Eine Formel ist dazu in der Lage, mehrere Zellen miteinander zu verknüpfen und den resultierenden Wert in der Zelle, in der die Formel steht, anzuzeigen.

Zum Aufstellen einer Formel stehen außer den mathematischen Funktionen des SuperBASIC noch weitere problemorientierte Funktionen zur Verfügung, z. B. Ermittlung der Reihe oder Spalte, in der man sich gerade befindet, oder Mittelwert, Maximum, Minimum, Summe und Anzahl der belegten Zellen eines bestimmten Bereiches, Verzinsung und Netto - Gegenwartswert ... Fortsetzung S.12

INHALTSÜBERSICHT

IMPRESSSUM:

HCR - HEIM + PERSONAL COMPUTER REPORT
Die Zeitung für Heim- und Personalcomputerinteressenten und -anwender.

HERAUSGEBER und VERLEGER:
Linus Wittich, Höhr-Grenzhausen

VERLAG:
Allpress-Verlagsgesellschaft mbH & Co.
D-5410 Höhr-Grenzhausen
Postfach, Kleine Schützenstraße 7
Telefon: (0 26 24) 50 99
Telex: 86 95 02 mgirm

Österreich:
Verlag + Druck Linus Wittich
A-6173 Oberperfuß-Berg, b. Innsbruck
Haus 165

Schweiz:
Otto Rys-Wellinger
Roosweldstraße 5
CH-8832 Wollerau

Chefredakteur: Hans Gerd Schneider

Graphik und Layout: Wolf-Dieter Hahn

Technische Leitung: Michael Wittich

Satz und Druck:
Verlag + Druck Linus Wittich
5410 Höhr-Grenzhausen

Mitarbeiter dieser Ausgabe:
R. Petruck, U. Haferland, D. Berner, F. Theis, J. Braun, H. Benson, O. Steinmeier, F. Thielen, T. Barndt, R. Bott, A. Lendlein, P. Eckhoff, F. Lorenz, R. Noppe, T. Schwarz, K. Weppler.

Vertrieb:
Verlagsunion
Friedrich-Bergius-Straße
6200 Wiesbaden
Anzeigenpreise:
z.Zt. ist die Anzeigenpreisliste Nr. 1/84 gültig.

Schlußtermine:
Als Schlußtermine gelten die Termine in unseren Mediaunterlagen.

Erscheinungsweise:
HCR - Heim + Personal Computer Report erscheint monatlich. Der Einzelverkaufspreis beträgt DM 1.50.

Abonnement:
Der Abonnementpreis für 12 Ausgaben beträgt 16.50 DM incl. Zustellung und MwSt.

Bankverbindungen:
Kreissparkasse Westerwald
Höhr-Grenzhausen
Konto-Nr.: 020-002770 (BLZ 570 051 01)

LESERBRIEFE

— Leserzuschriften —
werden von der Redaktion veröffentlicht ohne Rücksicht darauf, ob die darin zum Ausdruck gebrachten Ansichten mit der Meinung der Redaktion übereinstimmen. Die Redaktion behält sich sowohl die Veröffentlichung vor, wie auch sinnwahrende Kürzungen. Die Redaktion legt darauf Wert, daß die Zuschrift mit Namen und Anschrift des Einsenders veröffentlicht wird.

Betr.: Leserbrief von Rainer Paape in HCR 6/85

Sehr geehrte Redaktion,

da ich (fast) die gleichen Erfahrungen mit dem „Bund für Natur und Technik" gemacht habe, möchte ich auch dazu einen Leserbrief schreiben. In dem „Homecomputer" 8/84 wurde der Bund für Natur und Technik vorgestellt. Da mir die Leistungen, die der Club bietet, gefallen haben, bin ich in den Club eingetreten und habe mir auch gleich einen C 64 zum Preis von 529,- DM bestellt. Ein paar Tage später bekam ich von Ralf Seibel eine Auftragsbestätigung, mit der Bitte, 270,- DM anzuzahlen (der Preis war durch Sammelbestellungen so niedrig). Ich hatte die 270,- DM per Scheck beigelegt und danach längere Zeit nichts mehr von Ralf Seibel gehört. Nach schriftlicher Nachfrage, ob er den Scheck erhalten habe, bekam ich die Antwort „der Scheck sei bereits verbucht worden". Auch danach hatte ich weder den C 64 noch die von mir gezahlten 270,- DM erstattet bekommen. Auch meiner mehrfachen Aufforderung (u. a. per Einschreiben/Rückschein), mir den C 64 zu liefern oder den Betrag von 270,- DM zu erstatten, wurde nicht nachgekommen. Dann habe ich beim Amtsgericht Hamburg einen Mahnbescheid beantragt. Zwei Wochen später bekam ich einen Widerspruch gegen den Mahnbescheid. Den Widerspruch hat Ralf Seibel beim Amtsgericht Stuttgart gestellt und darin geschrieben, daß er von mir keine Verbuchung auf sein Konto feststellen konnte. Ralf Seibel hatte mir aber schriftlich bestätigt, daß mein Scheck bereits verbucht worden sei (Der Scheck ist auch von meinem Konto abgebucht worden). Von der Infozeitschrift „SYS", die einmal im Monat erscheinen soll, habe ich in der Zeit von 8/84 - 2/85 auch nur zwei Ausgaben bekommen.

Die Mitgliedschaft in dem Club lohnt sich nicht. Den Mitgliedsbeitrag von 36,- DM und 15,- DM für Porto kann man sich sparen. Für das Geld bekommt man aus Fachzeitschriften mehr und bessere Informationen.

Thomas Kausch
Wexstraße 35, 2000 Hamburg 36

Betr: Leserzuschriften

Ich möchte diese Gelegenheit nutzen, Ihnen ein dickes Lob für ihre Zeitschrift auszusprechen, nicht wegen des Preis - Leistungsverhältnisses, sondern auch wegen der Ausgewogenheit in den Berichten insbesonders aber auch in den Computertypen, die behandelt werden und für die Listings veröffentlicht werden. Bei allen anderen Zeitschriften ist hier eine zu starke Übermacht des C-64 zu verzeichnen, die dessen Marktposition nur untermauern und dafür sorgen, daß jeder C-64 Besitzer aus X-Serie, Grafik etc. Programm auswählen kann, während andere Computer vernachlässigt werden.

Ein Kritikpunkt sind aber die Listings, wenn ein Matrixdrucker oder, noch schlimmer, ein ZX-Druckerlisting noch verkleinert wird, wird das Lesen zur Plage, wenn ich doch davon ausgehen kann, daß die Redaktion über mindestens einen Computer der vier häufigsten Typen (C-64, CPC 464, ATARI, ZX-Spektrum) verfügt, so müßte es doch möglich sein, sich Programme auf einem Datenträger zusenden zu lassen und dann mit einem Typenrad oder Schönschreibdrucker auszudrucken.

Doch nun zu meinem eigentlichen Anliegen. Mit höchstem Interesse habe ich den fundierten Artikel über den MC-68 000 gelesen. Beim Geschwindigkeitsvergleich zwischen Z-80 C und MC-68 000 stutzte ich aber und schrieb ein entsprechendes Programm zur Zeitmessung auf meinem Rechner. Ich besitze einen CPC-464, der Gleitpunktoperationen mit 32 Bit Mantissen durchführt, also die Bedingung für einen objektiven Test erfüllt. Jedoch besitzt er nur einen Z-80 A, der mit 4 MHz, also der halben Taktfrequenz des Z-80 C läuft. Berücksichtigt man jedoch die Zeitverluste durch einen Interpreter, der ja bei jedem Schleifendurchlauf eine erneute Übersetzung vornimmt, so müßte er mehr als 2 mal langsamer sein als die angegebenen Werte. Jedoch gab es einige Überraschungen:

Z-80 (CPC-464)		7-80
Division:	2.94 ms	2.65 ms
Sinus:	14.33 ns	19 ms
EXP (x):	13.82 ms	14.5 ms

Die übrigen Werte stimmen mit der Tabelle überein, bei den Messungen ist die Zeit für die For-Next-Schleife schon abgezogen. Die Schleifen-Variable die gleichzeitig Argument ist, wurde von 0.01 bis 10.01 Step .0.01 gewählt sein. Bei den angegebenen Werten muß also die Software nicht optimal sein (Zeitmessung mit Timefunktion)

Nun noch eine Frage: Ich suche einen Basic-Compiler für den CPC, der aber alle Operationen des CPC's verarbeitet und nicht nur Ganzzahlvariablen. Hauptanwendung sind Programme mit technisch - wissenschaftlicher Ausrichtung oder mit vielen Stringoperationen. Kennen Sie einen solchen (Preis < 200 DM) ?
Ansonsten weiter so!

Bernd Leitenberger
Brühlstraße 22, 7302 Ostfildern 1

Betr. Leserbrief von Klaus Elschner

Sehr geehrte Damen und Herren, ich habe den Schneider CPC 464 und möchte damit gern eine Hardkopie vom Bildschirm machen.

Da ich jedoch keinen Schneiderdrucker, sondern einen Siemens Tintenstrahl-Drucker (PT 88) habe, ist es mir bis jetzt nicht möglich gewesen.

Ich wäre Ihnen dankbar, wenn Sie mir da weiterhelfen könnten.

Klaus Elschner
Wiesenstraße 5, 5090 Leverkusen 1

Anmerkung der Redaktion

In der nächsten Ausgabe von HCR-Heim und Personal-Computer Report werden wir ein 8-Bit Interface für den Schneider CPC als Listing bzw. Bauanleitung veröffentlichen. Mit diesem 8-Bit Interface ist es dann leichter möglich, Hardcopies vom Bildschirm zu machen.

Betr. Leserbrief von Thomas Schulz

Hallo HCR-Redaktion!
Ich bin im Besitz eines TI 99/4A und habe mich an Eurem Basic-Rätsel beteiligt. Um innerhalb der drei Zeilen-Grenze zu bleiben, ist das Programm in Extended Basic geschrieben. Es läßt sich aber leicht auf TI-Basic umschreiben.

Weiterhin möchte ich einmal gerne wissen, warum so wenig TI 99/4A - Artikel bei Euch erscheinen. Es gibt doch noch so viele TI - USER. Kann ich selbst auch einige Programme bei Euch veröffentlichen?

Ansonsten finde ich Eure Zeitung sehr gut, insbesonders die kritische Einstellung zum C 64, der ja hätte schon längst entthront werden müssen.

Weiterhin noch alles Gute und viel Erfolg mit Eurer Zeitung.

Thomas Schulz.
Kammerstraße 206, Bl. I, Zi.: 707, 41 Duisburg

Anmerkung der Redaktion

Wir möchten diesen Leserbrief einmal aufgreifen und zu dem Thema TI 99/4A Stellung nehmen. Wie die Leser von HCR-Heim- und Personal Computer Report wissen, werden in HCR sehr häufig noch Programme, Tips und Tricks für den TI 99 veröffentlicht. Das soll auch in Zukunft beibehalten werden. Aus diesem Grund sind wir jederzeit dankbar, wenn uns TI - User entsprechende Tips, Tricks und Programme zur Verfügung stellen.

Betr. Leserbrief von Paul Sattler

Sehr geehrte Damen und Herren, ich habe heute zum erstenmal Ihre Zeitung gekauft und gelesen. Leider ist mir beim Kauf nicht aufgefallen, daß auf dem Cover neben der Firma Apple und Epson auch die Fa. Schneider fett gedruckt ist.

Beim Lesen kam ich mir als C 64 User ziemlich verarscht vor. Ihr solltet Euch offen zu der CPC 464 bzw. 664 Zeitung bekennen, dann brauchte ich jetzt nicht meiner DM 1.50 nachzutrauern.

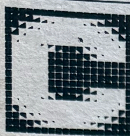
Dem Kollegen Petruck kann man nur raten, sein C 64er Wissen nicht vor die falschen Leser zu werfen.

Also Leute, keine Feigheit, bekennt Euch ruhig, der Trick mit dem Cover klappt eh nur einmal.

Paul Sattler
Riehler Gürtel 80, 5000 Köln 60

Anmerkung der Redaktion
Wir finden es schade, daß der Leser seiner DM 1.50 nachtrauert. Selbst wenn er nichts spezifisches für seinen C-64 gefunden hat, sind doch eine ganze Menge anderer nützlicher Informationen regelmäßig im HCR Heim- und Personal Computer Report zu lesen. Wer HCR länger liest, wird sicher wissen, daß die Vorwürfe, HCR sei eine reine Schneider Zeitung, nicht gerechtfertigt sind. Selbst das Heft, auf welches der Leser Bezug nimmt, die Ausgabe 7/85, enthält eine ganze Menge anderer Beiträge für Computer. Leider hat der Leser die Ausgabe 4 und 5 verpaßt, in denen die Programme Sprite Generator und Prosprite 1 für den Commodore 64 abgedruckt waren. Auch in Zukunft werden regelmäßig Listings und Artikel über den C-64 neben den anderen Computern vorhanden sein.

- Messe Köln -



Internationale Computer Show Köln
13. bis 16. Juni 1985
Computer für Beruf, Heim und Hobby

Talk - Show Software - Klau: Zusammenfassung der Resultate

Ein Trend zeichnet sich ab: Das Knacken von Programmen für Computer. Damit wollen viele Jugendliche ihr Taschengeld aufbessern. Andererseits wollen sie aber auch sichergehen, daß die Programme ihren Wünschen entsprechen.

Im Rahmen einer Podiumsdiskussion auf der Internationalen Computer - Show in Köln unter der Leitung von Richard Kerler (Redaktionsdirektor CHIP) kritisierten viele Jugendliche, daß ihnen die Industrie schlechte Programme verkauft. Deshalb würden sie sich die Programme von Freunden ausleihen, um sie zu kopieren.

Es gibt aber auch, wie Rechtsanwalt Freiherr von Grafenreuth als Teilnehmer der Diskussionsrunde ausführte, viele Jugendliche, die kommerziell Programme kopieren, um sich damit Geld zu verdienen. Gleichzeitig beklagt die Industrie die gewerbsmäßige Verbreitung von unerlaubten Kopien, bei denen sich einige Unternehmen Gewinne in Millionenhöhe sichern.

Die rechtliche Handhabung ist zwar einerseits im Urheberrecht geregelt, es fehlen aber noch Vereinfachungen wie sie z. B. im Bereich Musik oder in der Literatur bestehen. Mit anderen Worten: es gibt noch keine Verwertungsgesellschaft. Einheitliche Meinung aller Teilnehmer an der Podiumsdiskussion war, daß der Gesetzgeber schnellstens handeln müßte, um im Rahmen einer Verwertungsgesellschaft die Probleme zu lösen.

Erstaunlich großzügig zeigte sich Staatsanwalt Wolf aus Köln, der das Vorgehen gegen Jugendliche nicht zu eng sehen möchte. Für ihn besteht keine strafbare Handlung, wenn Programme im Freundeskreis kopiert werden. In diesem Fall würde er das Verfahren einstellen.

Allerdings dürften selbstverständlich nicht Auswüchse erfolgen, wie sie z. B. Rechtsanwalt von Grafenreuth zitierte: „Ein Computerclub bietet kopierte Programme an und sichert sich für diese illegal hergestellten Programme gleichzeitig noch einen Kopierschutz. Oder ein anderer Fall: Ein Unternehmer bietet die gesamte Software - Palette eines großen amerikanischen Herstellers zu dem Preis von DM 1000,- an, deren Wert in die Hunderttausende geht.

Als Fazit kann gelten: Die Jugendlichen müssen mehr aufgeklärt werden über die rechtlichen Auswirkungen des Kopierens von Software. Außerdem muß die Industrie alles daran setzen, „glaubwürdige Programme" auf den Markt zu bringen, um den Nutzern nicht das Gefühl zu geben, zu viel für schlechte Software zu bezahlen und sie so zu Raubkopien zu veranlassen. ■

Kritik und Randbemerkungen — von Patrick Schmitz —

An den Messekassen konnte sich der interessierte Besucher für 4,- DM einen ausführlichen Messekatalog kaufen. Dieser ist sehr übersichtlich gegliedert und hilft all jenen besonders, die den Stand einer bestimmten Firma suchen. Und das war gar nicht mal so einfach, denn die Ausstellungsfläche hat seit der ersten Show 1983 (damals noch US Computer Show) stark zugenommen!

Hat man nun endlich das Gebäude betreten, so bekommt man von einer hübschenDame auch gleich eine Plakette der C'85 angesteckt. Dies hat einen gut durchdachten Grund: An der Farbe der Plakette erkennen die Aussteller gleich, ob man als Besucher oder als Wiederverkäufer kommt, oder ob man beruflich unterwegs ist. Somit fällt es tatsächlich leichter, gleich die richtigen Kontakte zu knüpfen. Mit dem Personal der Aussteller konnte man zufrieden sein: Als Fachmann entpuppte sich der sympathische Herr von Honeywell Bull: Er kannte sich sogar mit den graphischen Fähigkeiten des legendären TI99/4A aus! Auch die attraktiven Damen von Apple zeigten nicht nur viel Bein, sondern auch die komfortable Bedienung des MacIntoshs mit der Mouse. Texas Instruments und Apricot glänzten mit sprechenden und hörenden Computern, und Schneider und Commodore zogen mit ihren neuen Geräten viele Interessenten an.

Bei soviel Information konnte man schon Appetit bekommen: kein Problem, denn Kaffee und Kuchen waren ebenso zu haben wie Süßes und Eis für die jungen Besucher. Viele Details und Nebensachen summieren sich so doch zu einem höchst angenehmen Messeaufenthalt, selbst wenn es wenig „weltbewegende" Neuigkeiten gab. Dies ist auch der einzige Kritikpunkt, für den die Organisatoren freilich nichts konnten: Im 16/32 bit -Bereich war leider nicht viel Neues zu vermerken, und das hätte wirklich nicht so sein müssen: Atari hat mit dem 520 ST schließlich DIE Computersensation der letzten Monate überhaupt geschaffen und war leider nicht auf der Messe präsent. Dies schmerzte umso mehr, da beimDATA BECKER - Stand nicht nur Jack Tramiel's Bild, sondern sogar schon Bücher zum Atari ST und ein 520 ST zu sehen waren. Dennoch möchte ich die Internationale Computer Show '85 in Köln auch aus der Sicht des Besuchers als vollenErfolg werten, und die - dünn gesäten - Mängel können ja behoben werden, denn vom 12.6. bis zum 15.6.1986 öffnet die C'86 ihre Tore ... man möchte fast wetten, dann auch mit Atari's ST - Reihe samt Zubehör und Software.... ■

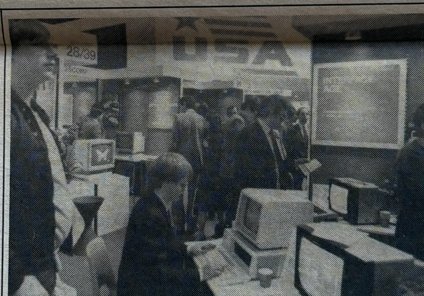
Neuer 16 bit-Rechner

Der neue GENIE 16 „C", in der Grundversion schon ausgerüstet mit 256 KRam, 2 Laufwerken mit je 360 K und hochauflösender Farb - Grafikkarte (wahlweise Monochrome - Karte) wurde vom Design her auf die bereits am Markt bekannten 8 - Bit TCS - Rechner abgestimmt.

Zur Grundausstattung gehören auch je eine serielle (V.24 bzw. RS 232 - C) und parallele (Standard - Centronics) Schnittstelle; ohne weitere Steckkarten kann auf der Hauptplatine bis auf 640 K erweitert werden.

Das Betriebssystem MS -DOS mit Basic - Interpreter gehört zur mitgelieferten Standardsoftware; Concurrent CP/M - 86 in der Version 3.1 kann optimal geliefert werden.

Der Verkaufspreis des GENIE 16 C beträgt 4995,- DM incl. Mehrwertsteuer; eine Harddiskversion mit 10 MB und zwei Laufwerken soll mit 7995,- DM incl. Mehrwertsteuer angeboten werden. Größere Festspeicher von 20 und 30 MB sind auf Anfrage erhältlich. TCS Computer GmbH, Kölnstraße 2-4, 5205 St. Augustin 2. ■



Auch die USA und andere Länder waren auf ICS vertreten. Alles in allem eine internationale Computershow

Neues Software - Programm verbindet Computer und Peripherie elektronisch

Mac Office stellt ein komplettes Bürosystem vor. Mit einem neuen steckerfertigen lokalen Netzwerk Apple - Talk lassen sich 32 Macintosh untereinander verbinden. Folgende Geräte gehören zu Mac Office: Macintosh, Apple - Talk, FileServer, Laserwriter sowie externe Laufwerke.

APPLE - Talk ist ein neues Software - Programm, welches Verbindung zwischen Computer und Peripherie elektronisch zusammenfügt. Über APPLE - Talk können 32 Macintosh - Benutzer untereinander kommunizieren, wobei nur 1 Laser - Writer notwendig ist. Ebenfalls können sich mit APPLE - Talk mehrere Nutzer den neuen Massenspeicher File - Server teilen. Mit APPLE - Talk ist auch eine Kommunikation mit Großrechner von EDV - Anlagen notwendig.

Eingebaute Zeichensätze geben dem Anwender den Vorteil, Zeichen und Grafik in Laser Auflösung zu erhalten. Die Qualität kommt der von Satzmaschinen gleich und ergibt sich aus der Auflösung von 300 Punkte/Inch gegenüber 70 Punkte/Inch bei Normaldruckern. Erreicht werden diese vielfältigen Leistungsmerkmale durch einen eingebauten Mikrocomputer mit 68000 - Prozessor, 1,5 MB RAM und 0,5 MB ROM. Die APPLE - Macintosh - Software ist mit dem Laser - Writer kompatibel, so daß z. B. professionelle Softwarepakete wie Jazz von Lotus oder die Software - Serien von Microsoft genutzt werden können. Eingebaute Schrifttypen: Helvetica. Times, Courier, Symbols.

Für die Anwendergruppen Handwerk, Rechtsanwälte, Wirtschaftsberater, Anlage-, Vermögensberater, Werbewirtschaft, Gewerbe, Notare, Unternehmensberater, Makler, Schule / Ausbildung, Forschung, Ärzte, Steuerberater, Architekten, Immobilien- und Wohnungswirtschaftsunternehmen, Führungskräfte.
Apple Computer GmbH, Ingolstädter Str. 20, 8000 München 20 ■

Der Commodore lernt „sehen"

Viele neue Möglichkeiten erschließen sich mit einem kleinen Gerät, das als Zusatzmodul zum C 64, zum 128 und zum SX angeboten wird.

Der VD 64 Digitizer überträgt Bilder von Kamera, Videorecorder oder Fernseher als digitalisierte Bilder auf den Computer - Bildschirm. Da das Gerät kein Standbild benötigt, kann auch von laufenden Bildern digitalisiert werden. Der Digitizer erstellt 3 Bilder pro Sekunde. Selbstverständlich ist auch ein Abspeichern der Bilder möglich. Das Bild wird in 4 Graustufen wiedergegeben und kann mit 4 Farben nachcoloriert werden; bei Verwendung als KOALA - Painters sind bis zu 16 verschiedene Farbstufen möglich.

Der VD 64 Digitizer ist in einem praktischen Kunststoffgehäuse eingebaut und kann ohne zusätzliche Stromversorgung direkt auf den User - Port des C 64 oder 128 oder SX aufgesteckt werden. Als Bildquelle können sowohl Videokamera als auch Videorecorder, über den Videoanschluß auch der Fernseher angeschlossen werden. Durch die gute Bildqualität und besondere Bedienungsfreundlichkeit ergeben sich vielfältige Anwendungsmöglichkeiten im Bereich der digitalen Bildverarbeitung.

Anwendungsbereiche: für Fans und Profis, für Film- und Video - Ergänzungen, Titelbilder, Übertragung von Bildern per Funk und Telefon, für technische Anwender, zur Auswertung von Bildveränderungen, für bestimmte technische Bereicherungen, für Druckereien und Werbefachleute, Alarmanlagen und vieles mehr.
Merkens EDV OHG, Fuchstanzstraße 6a, 6231 Schwalbach/Ts ■

Video - Digitizer

VIDEO DIGITIZER (Bildablageinterface Hard + Software) für alle gängigen Computersysteme zu äußerst niedrigen Preisen. Für die Systeme COMMODORE C 64/IBM + IBM Compatible FC's/APPLE COMPUTER / ATARI / SCHNEIDER etc. Mit den Systemen lassen sich VIDEO - SIGNALE aller Art einlesen, auf dem Schirm darstellen, auf Diskette ablegen und auf allen gängigen Farb- und schwarz - weiß - Druckern ausdrucken. Die Bilder lassen sich mit PAINT - Magig- / und / Koala - Pad weiterbearbeiten und ausdrucken. Ein Preisbeispiel für das Commodore Interface. Es kostet mit der Software 398,- DM. Computerperipherien Pitt-Joern Brockner, NNikolaistraße 2, 8000 München 40. ■

Netzwerk für PC-Rechner

MEMNET ist ein Netzwerk für PC - Rechner. 1 bis 16 Arbeitsplätze. Mit MEMNET ist es möglich - ohne Anpassung - alle Software unter MS - DOS 2.0 in Multi - user - System zu benutzen. Zum Schreiben ist die Datei geschlossen. Maximale Länge: 250 m zwischen zwei Arbeitsplätzen. Bus - Typ: in Stern oder Zeile - Konfiguration.

Entwicklungswerkzeug für PC

Entwicklungswerkzeug für PC unter MS - DOS 2.0 mit: - Basic (Neue ANSI Norma Prinzipien); Dateiverwaltung ; Maskengenerator; Multifensterverwaltung; Verwaltung des Farbbildschirmes und der Microsoft Maus. Für die professionelle Programmierung.
MEMSOFT GmbH, Dreieichstraße 59, 6000 Frankfurt a.M. 70. ■

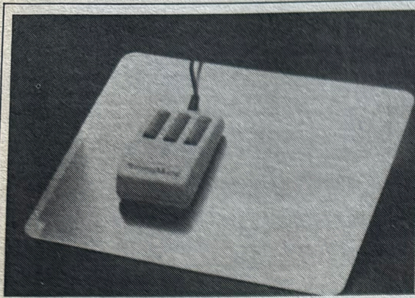
Der MSX - Express kommt in Fahrt

Gerade rechtzeitig zur C'85 in Köln gibt der Hersteller die Ankunft des MSX - Express bekannt. Es handelt sich dabei keineswegs um eine computerisierte Zugsteuerung, sondern um die neueste Bereicherung auf dem schnell wachsenden MSX - Markt. Erstmals bietet SVI mit dem MSX - Express ein Gerät an, das so komplett ausgestattet ist, daß lediglich noch ein Bildschirm angeschlossen werden muß, um den MSY - Express in Betrieb zu nehmen. Die folgenden Ausstattungsmerkmale werden sicherlich dazu beitragen, daß der MSX - Express schnell Fahrt aufnehmen wird:

- Schreibmaschinentastatur mit deutschen Umlauten
- eingebautes 3,5 Zoll - Diskettenlaufwerk einschließlich CP/M - Betriebssystem
- Anschlußmöglichkeit für 2. Diskettenlaufwerk
- eingebaute parallele Schnittstelle (centronics - kompatibel)
- eingebaute serielle Schnittstelle (RS232)
- MSX - Datenrecorder - Anschluß
- zwei MSX - Joystick - Anschlüsse
- eingebaute 80 - Zeichenkarte für den CP / M - Betrieb

Preis: DM 1498,-. Mit der ersten Auslieferung ist im Juli 85 zu rechnen. Bernd Jöllenbeck GmbH, Im Dorf 5, 2730 Weertzen. ■

Optoelektronische Maus



Unser Bild zeigt die neue Summa Mouse

SUMMAGRAPHICS stellt ein neues Produkt, die „Summa Mouse" vor, das entwickelt wurde, um dem Computerbenutzer eine billige Möglichkeit zu geben, den Bildschirmcursor zu kontrollieren.

Die preisgünstige und benutzerfreundliche „Maus" basiert auf dem Prinzip der Optoelektronik und ist mit einem Microprozessor ausgerüstet, um maximale Leistung erreichen zu können.

Die neue „Maus" von SUMMAGRAPHICS ist klein, leicht und robust und ist so konzipiert, daß sie bequem in die Hand paßt. Die drei Knöpfe des Cursors können ohne Blickkontakt manipuliert werden.

Die „SummaMouse" ist aus einigen sehr zuverlässigen Bausteinen gebaut und braucht keine Justierung. Ein Mikroprozessor errechnet die optimalen Arbeitskonditionen und kompensiert ggf. Ungenauigkeiten, z. B. aus Alterungsprozessen.

„SummaMouse" besitzt die einmalige Funktion „Autobaud", die seine Übertragungsgeschwindigkeit automatisch dem System anpaßt, mit dem sie verbunden ist. Sie kann von 300 auf 9600 Baud programmiert werden. Außerdem ist sie bereits mit einer V24- oder TTL - Schnittstelle versehen.

Im weiteren hat SUMMAGRAPHICS einen Digitalfilter eingebaut, der alle Fehler, die sich beim „teasing" produzieren können, vollständig ausschließt, ohne daß die Arbeitsgeschwindigkeit beeinträchtigt wird. ■

- Neu für Personalcomputer -

Software: Meßwerterfassung und -verarbeitung

Aufgrund spezieller Softwareprogramme und verbessertem Preis-Leistungsverhältnis finden immer mehr Personalcomputer Eingang in industrielle Anwendungsbereiche wie Versuch, Entwicklung, Produktion. Das Stuttgarter Institut für Softwareentwicklung Dr. techn. Braschel und Partner (IFB) stellte jetzt ein Softwareprogramm vor, nach dem auch komplizierte Meßwerterfassungs- und verarbeitungsvorgänge leicht und problemlos durchgeführt werden können. Das Programm basiert auf dem Siemens PC 16-11.

Das Institut für Softwareentwicklung in Stuttgart ist 1982 gegründet worden und befaßt sich ausschließlich mit der Entwicklung technisch orientierter Programmsysteme auf bekannten Rechenanlagen. Weiter erstellt es Mikroprozessorsysteme zur Meßdatenerfassung und -be- arbeitung sowie für Steuerungs- und Regelungsaufgaben. Neben der Softwareentwicklung ist das Hardwaredesign ein zusätzliches Arbeitsgebiet.

Das von den Softwarespezialisten erstellte neue Programm Unimess kann nach drei Hauptkriterien unterteilt werden: Vorbereitung des Meßwertvorhabens, Überwachung und Steuerung des Meßvorgangs sowie Nachbearbeitung und Dokumentation der Meßdaten. Die Nutzungsmöglichkeiten erstrecken sich von der Erstellung einer Meßgrößenliste unter Aufgabe der jeweils kennzeichnenden Stammdaten über Einblenden beliebig wählbarer Meßwertverläufe auf den Farbmonitor bis hin zum Erstellen eines Meßprotokolls.

Da die einzelnen Meßvorhaben sehr verschieden sind, werden auch äußerst unterschiedliche Anforderungen an die Leistungsfähigkeit eines Erfassungs- und Verarbeitungssystems gestellt. So sind für die Wahl der einzusetzenden Hardwarekonfiguration besonders entscheidend: Meßstellenzahl, Abtastrate, Umfang und Meßdatenverarbeitung. Um hier den vielen unterschiedlichen Anforderungen gerecht zu werden, müssen nicht nur modular aufgebaute Einzelgeräte, sondern zudem leistungsfähige Gesamtsysteme zur Verfügung stehen. Die IFB-Experten entschieden sich deshalb auch für den Siemens PC 16-11, weil dieser Personalcomputer mit Prozeßsteuerungen (Simatic 55) und Großrechnern voll kommunikationsfähig ist. Weiter überzeugt das Gerät durch universelle Einsetzbarkeit, hohen Bedienungskomfort und umfangreiche Dokumentationsmöglichkeiten. Im einzelnen bietet IFB folgende Hardwarevarianten an: Erfassungsbaugruppe als Stand-alone-Gerät, Personalcomputer und Prozeßadapter, Personalcomputer und Erfassungsbaugruppe.

Allgemein ausgedrückt: Der Siemens PC 16-11 kann mit dem neuen Programm überall dort eingesetzt werden, wo gemessen wird. So zum Beispiel im Motorenprüfstand bei Drehzahlen-, Abgas- und Schallmessungen. Ebenso in der Materialüberwachung zur Durchführung von Zugversuchen und Festigkeitsprüfungen. Ein weiteres Einsatzfeld bietet auch die Qualitätssicherung. Neben verschiedenen Forschungs-, Entwicklungs- und Prüfstellen der Industrie interessieren sich Hochschulen (Stuttgart, München etc.) und das Max-Planck-Institut für die neue PC-Software. ■

SOFTWARE-SCHUTZ

Die Firma GEFDA bietet neue Alternativen hinsichtlich des Kopierschutzes

GEFDA-SCHUTZ leistet einen großen Beitrag, lawinenartiges, unbefugtes Vervielfältigen einzudämmen. Der Schutz besteht aus zwei Hauptteilen.

Der erste Teil des Schutzes verhindert den normalen Ablauf aller DOS-Dienstprogramme, die zum Kopieren gedacht sind. Die DOS-Funktionen ‚validate' bzw. ‚collect' sind durch GEFDA-SCHUTZ lahmgelegt. Andere Kopierprogramme fallen generell in drei Gruppen, die einzeln erläutert sind.

Die erste und einfachste Gruppe Kopierprogramme richten sich nach dem Disketten-Verzeichnis. Diese erlauben dem Bediener, die Programme zu wählen, die er kopieren möchte. Je nach dem, welche Programme gewählt werden, gerät das Kopierprogramm in eine endlose Schleife oder es überträgt absoluten Unsinn ohne daß das Kopierprogramm etwas davon merkt.

Kopierprogramme aus der zweiten Gruppe richten sich nach dem ‚BAM' (block availability map). All diese Programme übertragen ebenfalls absoluten Unsinn. Auch dann, wenn eine nachfolgende Prüfung stattfindet, bestätigen solche Kopierprogramme, daß alles in Ordnung sei.

Kopierprogramme aus der dritten Gruppe, die die ganze Diskette Byte für Byte, egal was kommt, übertragen, melden ‚read error'. Wenn diese trotzdem weiterfahren, erzeugen sie zwar ein Duplikat, dieses ist aber mit einer Ausnahme nicht lauffähig, und zwar die Disk-Doktor-Programme.

Der zweite Teil des GEFDA-SCHUTZES wird in das zu schützende Programm eingebaut. Dieser Teil besteht aus etwa 30 Anweisungen in der Programmiersprache BASIC.

Unter anderem stellen diese Befehle fest, ob die obigen Schutzmaßnahmen auf der Diskette noch vorhanden sind. Durch diesen Teil stellt der Programmverfasser fest, ob sein Programm von der originalen oder von einer fremden Diskette geladen wurde. Je nachdem, ergreift er passende Maßnahmen.

GEFDA-SCHUTZ kann so angewendet werden, daß ein einziges, geschütztes Programm von allen GEFDA geschützten Disketten lauffähig ist oder aber *nur von einer einzigen.* ■

APPLE

Macintosh 512 KB mit 20 MB - Festplatte

Ab Herbst dieses Jahres wird der Macintosh 512 KB mit einer externen 20 MB Festplatte geliefert. Wie die Apple Computer Inc., Cupertino/Kalifornien, jetzt ankündigt, erweitert die neue Macintosh - Konfiguration das modular aufgebaute „Macintosh Office" nach oben.

Die neue Festplatte ist für den Macintosh als individueller Arbeitsplatz entwickelt worden und kann 50 mal mehr speichern als eine 400 KB Floppy - Disk. Die externe 20 MB - Festplatte ergänzt den auf der Hannover - Messe in diesem Jahr angekündigten FileServer, der innerhalb des neuen Apple - Netzwerkes „AppleTalk" ein Hauptelement für den Multiuser - Betrieb des „Macintosh Office" ist.

Mit der Leistungserweiterung des neuen Macintosh unterstreicht Apple die künftige Konzeption einer modular aufgebauten Macintosh - Linie, die dem „Persönlichen Computern" im professionellen Einsatz auch bei zunehmenden Leistungsansprüchen gerecht wird. Weitere Spezifikationen des Macintosh 512 KB mit 20 MB - Festplatte sowie eine Preisangabe wird die Apple - Zentrale in Cupertino im Herbst bekanntgeben.

Auslaufen wird die Modell - Reihe Macintosh XL (Lisa), da nach der Apple - Produktstrategie künftig nicht mehr zwei kommerzielle Computersysteme, sondern nur noch die Macintosh - Linie neben der Apple II - Familie verfolgt wird. Apple wird Service und Support für die „Lisa" weiterhin aufrechterhalten. Mit dem Software - Programm Mac Works laufen die meisten Softwarepakete für den Macintosh auch auf der „Lisa". Insgesamt werden derzeit mehr als 500 Software - Programme für den Macintosh angeboten.

Anwendern des Bürosystem - Paketes 7/7 soll es ermöglicht werden, ihre Daten - Files zu konvertieren, um diese auf Macintosh weiterverarbeiten zu können.

Außerdem läßt sich die „Lisa" in das Apple Talk - Netz einbinden und an dem neuen Apple LaserWriter anschließen. ■

Frankfurter MICRO - COMPUTER '86 vom 29. Januar bis 2. Februar

Mehr Ausstellungsfläche - Fachcharakter verstärkt

1986 findet die MICRO-Computer - Internationale Frankfurter Mikrocomputer-Messe von Mittwoch, 29. Januar bis Sonntag, 2. Februar, statt. Das beschloß der Fachbeirat nach dem großen Erfolg der Erstveranstaltung MICRO-COMPUTER '85 in Frankfurt.

1986 wird zusätzlich zur Ebene 0 auch die Ebene 1 der Halle 4 eingesetzt. Die Fachbesucherzeiten werden erweitert: Mittwoch, 29. Januar und Donnerstagvormittag, 30. Januar 1986 sind exklusiv für den Fachbesuch reserviert.

Zur MICRO-COMPUTER '86 wird erneut ein umfangreiches, verbreitertes Rahmenprogramm für spezielle Zielgruppen wie Einzelhandel, Handwerk und Gesundheitswesen angeboten.

ASSEMBLER FÜR DEN QL

Der QL Assembler, ein Maschinencode - Dienstprogramm für den Sinclair QL Personal Computer, ist jetzt im Handel verfügbar.

Das von GST Computer System geschriebene Programm wurde in erster Linie für den technisch orientierten QL Anwender geschrieben, der die anspruchsvolle 32 Bit Architektur und das hohe Leistungspotential des QL vollständig ausschöpfen möchte.

Der nicht verschiebbare, mit dem Motorola Format kompatible 68000 Assembler konvertiert in M 68000 erstellte Quelldateien in QL - kompatible Binärdateien.

In das QL Assemblerprogramm integriert ist das von Metacomco entwickelte Sinclair Multitasking Bildschirm - Editor - Programm. Editorprogramm wie auch Assemblerprogramm können gleichzeitig mit Super-BASIC eingesetzt werden. Der Anwender kann ganz leicht zwischen den drei Programmen umschalten. Der Assembler kostet 178 DM. Für weitere Informationen: Sinclair Research Ltd., Niederlassung Deutschland, Hoehestr. 46 - 48, 6380 Bad Homburg. ■



Mit dem neuen Assembler werden zahlreiche Türen für Programmierer geöffnet

Schachprogramm für den — SINCLAIR QL —



Schach Matt mit Sinclair's QL

Neu auf dem Markt ist „QL Chess" für den Sinclair QL Personal Computer. „QL Chess" ist ein anspruchsvolles Schachprogramm in dreidimensionaler Grafik. Geschrieben wurde „QL Chess" von Psion Ltd. zusammen mit Richard Lang. Das Schachprogramm hat 28 Schwierigkeitsgrade und fast 4000 Züge. Erhältlich im Handel für 98 DM. Für weitere Informationen: Sinclair Research Ltd., Niederlassung Deutschland, Hoehestr. 46 - 48, 6380 Bad Homburg. ■

Raubcopy zerstört Computer!

Warum und wieso Raubcopien überhaupt existieren, dürfte wohl jedem klar sein. Der Preis eines Original - Programms ist effektiv zu hoch. Einmal ganz abgesehen von den Schrottprogrammen, welche sich hier und da auch noch auf dem Markt für dreißig bis sechzig Mark herumtummeln und auf einen ,,Dummen'' warten, der sie kauft. Ob sich hiervon eine Raubcopy für einen ,,Programmsammler'' lohnt, sei erst einmal dahingestellt.

Von R. Petruck

Also steigen die Raubkopierer nach einer gewissen Zeit auf qualitativ hochwertige Programme um. Jedoch spätestens hier steigt mein Blutdruck schneller an! Hat sich einer die Schreiberitis mit viel ,,Ach und Krach'' einigermaßen selbst beigebracht, weil keiner existierte, von dem man es lernen konnte, dem ist bekannt, mit wieviel Arbeit so ein Programm zu schreiben verbunden ist. Das bisher stärkste Stück passierte mir in einem Kaufhaus. Ich selbst besaß nur einen VC - 20 und den C - 64. Da es mich aber schon seit längerer Zeit reizt, einmal auf einem SCHNEIDER zu klimpern, ging ich in ein Warenhaus und startete meine ersten Gehversuche. Ich hatte schließlich keinen ,,Bock'', mir diese Kiste zu kaufen, um festzustellen, ob das Programmieren mit dem tapferen Schneiderlein auch Spaß macht. Also gings in der Computerabteilung ran an die



Die meisten Raubkopien existieren ohne Zweifel von C 64 Programmen

Profis mit 14 Jahren

Bulletten. Ich versuchte mein Glück mit Menü's und Eingabe - Masken. Nach ca. vier oder fünf Stunden war auch schon etwas Brauchbares auf dem Monitor zu sehen. Als ich so auf dem Bildschirm mein Werk betrachte, sehe ich aus den Augenwinkeln links neben mir einen Jungen stehen, vielleicht vierzehn oder fünfzehn Jahre alt, mit einer Audiocassette in der Hand. Da ich mir schon vorstellen konnte, was er von mir wollte, stieg mein Adrinalinspiegel ganz gehörig an, und bevor ich mir der Wahl meiner Worte bewußt wurde, floß auch schon aus meinem Munde: ,,Was willst Du denn?'' Seine Augen strahlten und mit einer Wahnsinnsbegeisterung fragte er mich: ,,Kann ich me dat kopiere?'' Das ist doch nicht möglich, denke ich, Du sitzt hier fast fünf Stunden, und er will das mal eben, nur so zum Spaß, kopieren. Aber bevor ich mich weiter aufrege, will ich's mal auf die Diplomatische versuchen und frage ihn: ,,Du schreibst wohl selbst keine Programme, was?'' ,,Enä kann ich net'', war seine abwertende Antwort. Als ich abschließend zu ihm sagte, er möge dies erst einmal lernen, dann könnte er auch eine Copy von mir haben, zwitscherte er beleidigt von dannen.

Mit anderen Worten, auch dieser Freak ist sich überhaupt nicht bewußt, wo Software - Klau und Raubcopy anfangen bzw. enden. Sehr viele werden mit meiner Meinung nicht übereinstimmen, aber ich behaupte, daß der ,,Killerschutz'' von Dirk P. Hassinger in einige Programme einfach hineingehört. Wenn man bedenkt, daß ein Programmierer manchmal bis zu einem Jahr an einem Programm arbeitet, bis es verkaufsreif ist. Wo bleibt der Lohn, wenn man solche Programme in vier Minuten

Killerschutz

kopiert und vielleicht auch noch vertreibt? Was macht nun D.P. Hassinger's Killerschutz? Ganz einfach: Bei Kopierversuchen sagen drei Chip's im 64iger Aufwiedersehen, von nun an ohne uns DREI!! Reparaturkosten: ca. 150,- DM. Wie funktioniert nun dieser Killerschutz. Im C - 64 fließt Strom, wie jeder weiß und eine Menge kleiner Schalter befinden sich auch dort drin. Ist man nun in der Lage, alle Schalter gleichzeitig zu schließen, so hat man den gleichen Effekt als wenn man mit einer Schere in die Steckdose geht! Der Strom hat dann eben ausgeflossen. Dank eines kräftigen kurzen Ende, aus der Traum. Sicherlich ist es nur eine Frage der Zeit, bis auch dieser Killerschutz geknackt wird. Jedoch wird einem erst einmal die Lust am Kopieren vermiest, denn ein einziger Kopierversuch genügt, um den Rechner zu zerstören. Zur Zeit kann allerdings noch fleißig weiter kopiert werden, denn DATA - BECKER zögert noch mit dem Vertrieb, weil die Sache juristisch noch etwas zwiespältig ist. Gibt man aber das Okay, sind eine Woche später die ersten Originale mit Killerschutz im Handel. Außerdem spricht Dirk P. Hassinger an alle Cracker eine Warnung! Wenn der Schutz nicht vertrieben werden sollte, bringt er eventuell selbst einige präparierte Kopien im Umlauf. Denn auch er sieht nicht ein, daß man permanent geistig beklaut wird. Es ist nur schade, daß diese Auffassung nur von den Leutchen geteilt wird, die selbst Programme schreiben und somit wissen, welche Arbeit dahinter steckt. Das traurige an den Raubkopien ist, daß sie oft nur gesammelt werden und nach ein oder zweimaligem Ausprobieren in irgendeiner Ecke liegen und Staub ansetzen. So hat eben jedes Ding zwei Seiten. ■

Physik per Computer

Teil 3: Der schiefe Wurf

Von F. Theis

3.1.1. ohne Berücksichtigung des Luftwiderstandes Allgemeine Betrachtungen

Der schiefe Wurf ist eine zusammengesetzte Bewegung aus
- dem freien Fall (gleichförmig beschleunigt)
und
- einer Bewegung mit konstanter Geschwindigkeit (V_0).

Die Wurfbahnen ergeben sogenannte Wurfparabeln, die sich nach folgenden Gleichungen berechnen lassen:

- Wurfhöhe $H = \frac{(V_0 \cdot \sin\ )^2}{2g}$
- Flugzeit $T = \frac{2 \cdot V_0 \cdot \sin}{g}$
- Wurfweite $W = \frac{V_0^2 \cdot \sin 2}{g}$

Die Größen H,T und W sind Eckwerte unserer Betrachtungen und könnten auch leicht mit dem Taschenrechner berechnet werden. Wesentlich wichtiger für die graphische Darstellung ist die Berechnung der Punkte P(x,y) auf der Bahnkurve.

Da sich die Geschwindigkeiten überlagern, wirkt auf die y-Komponente die gleichförmige Gechwindigkeitszunahme durch den freien Fall. Wir erhalten

$x = V_0 \cdot t(x) \cdot \cos$

$y = V_0 \cdot t(x) \cdot \sin \quad - g \cdot t^2/2$

Somit erhalten wir die Punkte, die man benötigt, um die Kurve plotten zu lassen.

3.1.2. Programmbeschreibung

Das Programm berechnet die Größen H, T und W beim schiefen Wurf eines beliebigen Körpers und plottet die Bahnkurven. Alle Größen werden für je drei verschiedene Abwurfgeschwindigkeiten und Abwurfwinkel berechnet. Weiterhin kann der Benutzer entscheiden, ob er nur die Bildschirmdarstellung oder Plottendarstellung wünscht. Während das Programm bearbeitet wird (Rechenphase) erscheint auf dem Monitor die in Abb. 1. dargestellte Anzeige.

```
*************************
*     Bitte warten!!    *
*************************
```

Die Darstellung der Bahnkurven erfolgt so, daß bei fester Geschwindigkeit V_0 die verschiedenen Abwurfwinkel berechnet und geplottet werden.
Für die Geschwindigkeiten
V_0 = 1oo m/s , 125 m/s und 15o m/s
und die Winkel
= 3o o , 45 o und 85 o
sind die Berechnung im Beispiel auf S. 9/1o (Abb.2. - 5.) durchgeführt worden.

3.1.3. Abschließende Bemerkung

Das Programm zeigt deutlich, wie nützlich der Homecomputer sein kann, wenn es darum geht, einen physikalischen Sachverhalt einleuchtend darzustellen.

```
100 REM*********************************
110 REM
120 REM   S C H I E F E R  W U R F
130 REM   ------------------------
140 REM
150 REM     Programmnane "WURF"
160 REM
170 REM  Das Programm berechnet die
180 REM
190 REM      * Wurfweite
200 REM      * Wurfhoehe
210 REM      * Flugzeit
220 REM
230 REM  und plottet die Flugbahn.
240 REM
250 REM  © 1985 by Friedhelm Theis
260 REM
270 REM*********************************
280 CLS
290 REM  1.Beschreibung
300 REM  --------------
310 REM
320 PRINT:PRINTTAB(8)"S c h i e f e r  W
 u r f"
330 PRINTTAB(8)"========================
"
340 PRINTTAB(2)"Das Programm berechnet f
";CHR$(173);"r drei ver-"
350 PRINTTAB(2)"schiedene Abschu";CHR$(1
74);"winkel PHI(1..3) und"
360 PRINTTAB(2)"Abschu";CHR$(174);"gesch
windigkeiten V(1..3) die":PRINT
370 PRINTTAB(2)"* Wurfweite"
380 PRINTTAB(2)"* Wurfh";CHR$(186);"he"
390 PRINTTAB(2)"* Flugzeit":PRINT
400 PRINTTAB(2)"und plottet die Flugbahn
":PRINT
410 PRINTTAB(2)"Eingabe"
420 PRINTTAB(2)"-------"
430 PRINTTAB(2)"* Abschu";CHR$(174);"ges
chw: V(1) ... V(3)"
440 PRINTTAB(2)"* Abschu";CHR$(174);"win
kel PHI(1) ..PHI(3)":PRINT
450 PRINTTAB(2)"Ausgabe"
460 PRINTTAB(2)"-------"
470 PRINTTAB(2)"* Tabelle mit Weite, H";
CHR$(186);"he und Zeit"
480 PRINTTAB(2)"* Wurfparabeln":PRINT
490 PRINTTAB(5)"V E R S T A N D E N ? J/
N ↓"
500 GET A$
510 IF A$<>"J" THEN 500
520 CLS
530 REM  2.Eingabe und Berechnung
540 REM  ------------------------
550 REM
560 DIM V(3),PHI(3)
570 DIM X(101,3),Y(101,3)
580 DIM H(3,3),T(3,3),W(3,3)

590 PRINT:PRINT:PRINTTAB(5)"Abschu";CHR$
(174);"geschw.",
600 INPUT"V(1) ... ?";V(1)
610 PRINTTAB(19)" ";
620 INPUT"V(2) ... ?";V(2)
630 PRINTTAB(19)" ";
640 INPUT"V(3) ... ?";V(3)
650 PRINT:PRINTTAB(5)"Abschu";CHR$(174);
"winkel ",
660 INPUT"PHI(1).. ?";PHI(1)
670 PRINTTAB(19)" ";
680 INPUT"PHI(2).. ?";PHI(2)
690 PRINTTAB(19)" ";
700 INPUT"PHI(3).. ?";PHI(3)
710 PRINT:PRINT:PRINTTAB(5)"Soll geplott
et werden ?"
720 PRINT:PRINTTAB(8)"Ja ..... 1"
730 PRINTTAB(8)"Nein.... 2"
740 GET P1
750 IF (P1<1)+(P1>2) THEN 740
760 CLS
770 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
780 PRINTTAB(5)"************************
****":PRINT
790 PRINTTAB(8)"B i t t e  w a r t e n"
800 PRINT:PRINTTAB(5)"*****************
**********":PRINT
```

Fortsetzung auf Seite 10

—PASCAL—

für BASIC - Kenner

Teil 9

Wer das bisher Gelernte überschaut, wird feststellen, daß bei Pascal viel mehr von vornherein festzulegen ist, als bei BASIC: Variablen müssen vorher deklariert werden, die maximale Länge von Zeichenketten muß vorher festgelegt werden ... So wird es umso mehr überraschen, daß es in Pascal sogar etwas gibt, wobei man weniger als in BASIC vorher festzulegen hat: die Listen

Von Dieter Berner

```
10 PROGRAM KLEINSCHREIBUNG(INPUT,OUTPUT);
20 VAR C:CHAR;
30 BEGIN
40 REPEAT
50 GET C;
60 IF ORD(C) IN [193..218]
70 THEN WRITE (CHR(ORD(C)-128));
80 ELSE WRITE (C);
90 UNTIL C="←";
100 END.

READY.
```

Auflösung aus Heft 7/85

Wie Sie sehen, ist die Lösung der letzten Aufgabe sehr kurz ausgefallen. (Nachdem das Programm mit RUN gestartet wurde, erscheint der Text auf dem Bildschirm in Kleinbuchstaben, auch wenn die SHIFT - Taste gedrückt wird. Programmabbruch durch Drücken der ,,Pfeil nach links'' - Taste. Vor demStart den Computer in den Kleinschreib - Modus versetzen!) Hier wird der Vorteil des Arbeitens mit Mengen deutlich: In kurzer Form lassen sich Vergleiche mit einer Reihe von Elementen durchführen, ohne wie in BASIC Zählschleifen einzusetzen. BASIC ließe hier auch eine Entscheidung (>192 und <229) zu, aber spätestens dann, wenn die Menge aus mehreren Teilmengen und vielleicht noch einigen einzelnen definierten Elementen besteht, wird die Überlegenheit der Mengenstruktur bei manchen Programmen deutlich.

Wer das bisher Gelernte überschaut, wird feststellen, daß bei Pascal viel mehr von vornherein festzulegen ist, als bei BASIC: Variablen müssen vorher deklariert werden, die maximale Länge von Zeichenketten muß vorher festgelegt werden ... So wird es umso mehr überraschen, daß es in Pascal sogar etwas gibt, wobei man weniger als in BASIC vorher festzulegen hat: die **Listen**. (Ich wähle hier den bei ,,Pascal 64'' verwendeten Begriff. Andere Autoren sprechen dabei nur vom ,,dynamischen Speichern'' und verwenden das Wort ,,Liste'' im herkömmlichen Sinne).

Bei dieser neuen Struktur ist es am Anfang unwichtig, aus wievielen Elementen sie schließlich bestehen wird. Solange der Speicherplatz des Rechners ausreicht, wird bei einer Neueingabe den Variablen Speicherraum zugeteilt. Nicht einmal die Reihenfolge der Elemente muß von vornherein festliegen. Jedes Listenelement hat zumindest einen POINTER (auf deutsch ,,Zeiger''), der in der Regel auf das nächste Element zeigt. Außerdem enthält jedes Listenelement mindestens eine andere Variable, um derentwillen es überhaupt gespeichert wurde. Dabei kann es sich beispielsweise um Namen, Wohnorte, Geburtsdaten, Preise, Koordinaten, Gewichte ... handeln. Um hier Variablen der verschiedenen Datentypen zusammenfassen zu können (und auch, um den POINTER unterbringen zu können) ist es sinnvoll, Listenelemente als RECORDs zu deklarieren.

Listen stellen eine Struktur dar, die in BASIC nichts Gleichwertiges hat. (Das heißt nicht, daß man die gleichen Probleme nicht auch in BASIC lösen könnte!) Deshalb ist es auch in diesem Fall nicht sinnvoll, dem Pascal - Programm ein BASIC Äquivalent gegenüberzustellen.

Gerade weil das ,,dynamische Speichern'' dem BASIC - Programmierer so fremd ist, möchte ich versuchen, den Sachzusammenhang mit einem Spiel zu klären: (Die Entsprechungen der Pascal - Listen stehen jeweils in Klammern!)

Es gibt Spiele - als Geländespiele oder auch als Autorallyes - bei denen die Teilnehmer beim Start nur ihr nächstes Ziel kennen. Dort angekommen, erhalten sie neue Informationen (die oben erwähnten Variablen!), welche Aufgaben sie zu erledigen haben: Außerdem bekommen sie einen Hinweis (POINTER!), wo die nächste Station liegt.

Vergleiche ,,hinken'' immer, aber bei aller Unterschiedlichkeit der Themen zeigt das Beispiel doch alles Wesentliche. Es zeigt auch, wie die Veranstalter des Spiels eine Zwischenstation, die sie vergessen haben, nachträglich noch einbauen können:

Vorher:
Station A - Wegweiser: gehe nach B
Station B - Wegweiser: gehe nach C.

Einzubauende Zwischenstation: X

Nachher:
Station A - Wegweiser: gehe nach X
Station X - Wegweiser: gehe nach B
Station B - Wegweiser - gehe nach C

Man muß also beim Einfügen lediglich den Pointer der vorangehenden Station ändern und den bisherigen Pointer der Zwischenstation zuweisen!
Zum Gelingen des Spiels fehlt jetzt eigentlich nur etwas ganz Banales: Die Teilnehmer müssen wissen, wo der Start des Spiels ist, denn sonst ist alles weitere hoffnungslos. Also müssen wir außer den Pointern der Listenelemente gleich noch einen weiteren vereinbaren, der die Startposition kennzeichnet. Außerdem sollte bei der letzten Station zu erkennen sein, daß man am Ziel angekommen ist, und daß weiteres Suchen also zwecklos wäre. Das erreicht man bei Pascal, indem man dem Pointer des jeweils letzten Listenelements den Wert NIL (not in list) zuweist. Bevor das erste Listenelement eingegeben wird, muß verständlicherweise die Startvariable diesen Wert zugewiesen bekommen.

Es gibt verschiedene Wege, die Listenelemente miteinander zu verknüpfen. Der einfachste ist folgender:

Vor der Eingabe des ersten Elements:

START: = NTL;

Eingabe des 1. Elements:

NEW (A) (Zuteilen von Speicherraum) A = 1
Pointer des 1. Elements: = bisherige Startadresse
in diesem Fall NIL - dadurch wird beim Durchsuchen der Liste mit diesem Element aufgehört)
START: = A (in diesem Fall 1, also ist die neue Startadresse 1!)

Eingabe des 2. Elements:
NEW (A) (Zuteilen von Speicherraum) A = 2

Pointer des 2. Elements: = bisherige Startadresse (in diesem Fall 1)

START: = A (in diesem Fall 2, also ist die neue Startadresse 2)

Das Vorgehen ist auf den ersten Blick etwas ungewöhnlich: Die neuen Elemente werden vor das bisher erste Element gesetzt. Damit ändert sich die Startadresse, und beim Durchsuchen der Liste wird mit dem zuletzt eingegebenen Element begonnen. Das zuerst eingegebene Element hat als Pointervariable NIL übernommen und bildet damit das Schlußglied in der Kette!

Eine schematische Darstellung mag den Zusammenhang verdeutlichen:

vor d. Eingabe		Eing.d.1.Elem.		Eing.d.2.Elem.		Eing.d.3.Elem.	
Adr.	Nächstes	Adr.	Nächstes	Adr.	Nächstes	Adr.	Nächstes
		1	NIL	1	NIL	1	NIL
				2	1	2	1
						3	2
Startadresse							
NIL		1		2		3	

Die Adressen der Speicherplätze wurden in diesen Beispielen vereinfachend mit 1, 2, 3 ... angegeben, was aber den prinzipiellen Zusammenhang trifft.

STADT (Adresse)	
STADT↑ (Gesamtheit der Inhalte des Listenelements)	STADT↑.NAECHSTE (Pointer, auf die folgende Adresse gerichtet)
	STADT↑.NAME (Array of CHAR)
	STADT↑.EINWOHNER (Integervariable: Einwohnerzahl)
	STADT↑.STAAT (CHAR-Variable, verschlüsselte Angabe des Staates)

Während in manchen Pascal - Implementationen die POINTER als POINTER - Variablen deklariert werden müssen, hat man bei ,,Pascal 64'' die Wahl, ob man als Variablentyp POINTER oder INTEGER angeben will:

Am Ablauf des Programms ändert beides nichts!

Die Bereitstellung von Speicherplatz erreicht man durch den Befehl NEW (X), wobei für X jeweils der Name des RECORDs einzusetzen ist.

Wie üblich, so hat auch in den Listen jeder RECORD einen Namen - im folgenden wird der RECORD - Name mit ,,STADT'' angenommen. (Es könnte sich etwa um ein Programm zur statistischen Aufwertung von Einwohnerzahlen handeln.)

Das Schema zeigt den prinzipiellen Aufbau eines Listenelements: (Die in Klammern gesetzten Angaben stellen nur Erläuterungen dar.)

Doch jetzt zur Praxis: Das Beispiel ist diesmal recht ausführlich, denn es soll außer der Eingabe der Listenelemente zeigen, wie man menügesteuert die Liste nach unterschiedlichen Gesichtspunkten durchsuchen kann.

(Zu Übungszwecken könnten Sie die WHILE - Schleifen durch REPEAT - UNTIL - Schleifen ersetzen und die drei IF... THEN... - Entscheidungen durch eine CASE...OF - Struktur!)

Problem

Zunächst sollen die Daten für den Lagerbestand einer Firma angegeben werden. Für jeden Artikel werden Artikelnummer, Bezeichnung und Preis eingetippt. Dann kann herausgesucht werden:

Wenn nach einem bestimmten Artikel gesucht wird, dann sollen Bestellnummer und Preis ermittelt werden. Falls nur die Bestellnummer bekannt ist, dann sollen Artikelbezeichnung und Preis ausgegeben werden.

Bisher wurde stillschweigend vorausgesetzt, daß jede Liste Anfang und Ende haben müßte, und daß jedes Listenelement einen und nur einen Nachfolger hätte. Das muß aber überhaupt nicht so sein: Eine Liste kann eine Kreisstruktur erhalten, wenn der POINTER des letzten Listenelements wieder auf das erste Element verweist. Andererseits brauchen Sie nur an die Urlaubsreise zu denken, um zu wissen, daß man in (fast) jedem Ort mehrere Entscheidungsmöglichkeiten hat, welches die nächste Station sein soll. Abhängig von dieser Entscheidung werden sich für die weitere Reise völlig verschiedene Routen ergeben. Diese Möglichkeit, die auch als ,,Entscheidungsbaum'' bezeichnet wird (ein Element hat mehr als einen Nachfolger) steht Ihnen auch bei der Verarbeitung von Pascal - Listen offen, und es ist nicht die einzige Möglichkeit ,,für Fortgeschrittene''. Solche Betrachtungen würden aber den Rahmen dieser Reihe sprengen, und so kann ich Sie nur auf die Fachliteratur verweisen.

Auch wenn die verschiedenen Pascal - Implementationen einander ähnlicher sind als unter-

Software Top-Twenty C 64

1. Rocky Horror Show
2. Hexenküche
3. Way of Exploding F.
4. Jump Jet
5. Superstar Challenge
6. Super Huey
7. Eureka (deutsch)
8. Summer Games
9. Flight Simulator
10. Amazone
11. Impossible Mission
12. Talladega
13. Operation Swordfish
14. Dummy Race
15. Music Studio
16. Dambusters
17. Pitstop 2
18. Ghostbusters
19. Slapshot
20. F 15 - Strike eagle

Software Top-Twenty CPC 464

1. Flight Path 737
2. Ghostbusters
3. Top-Data
4. Money Molch
5. Time
6. Grand Prix Driver
7. Nibbler
8. Mr. Pingo
9. Quicktext
10. Multi-Text
11. Paintbox
12. Figher Pilot
13. Beach Head
14. Minder
15. Superpipeline 2
16. Frankn'stein
17. Tascopy
18. Jet Boot Jack
19. Sorcery
20. 3d--Time-Trek

Index Rushware

```
10 PROGRAM LAGER(INPUT,OUTPUT);
20 VAR A,START:↑RECORD
30 ;                   NAECHSTES:POINTER;
40 ;                   NR:INTEGER;
50 ;                   ARTIKEL:ARRAY[1..12] OF CHAR;
60 ;                   PREIS:REAL;
70 ;              END;
80 GESUCHT:INTEGER; BEZEICHNUNG:ARRAY[1..12] OF CHAR;
90 MODUS,WEITER:CHAR;
100 BEGIN
110 MODUS:= "A"; START:= NIL;
120 WHILE MODUS<>"D" DO
130 BEGIN
140 WRITE ("   B I T T E   W A E H L E N   S I E !   ");
150 WRITE ("        A DATENEINGABE                   ");
160 WRITE ("        B SUCHEN (ARTIKEL BEKANNT)       ");
170 WRITE ("        C SUCHEN (ARTIKELNUMMER BEKANNT) ");
180 WRITELN ("       D ENDE                          ");
190 READLN (MODUS);
200 IF MODUS="A" THEN BEGIN
210 ;    NEW (A);
220 ;    WRITE ("ARTIKELNUMMER: "); READLN (A↑.NR);
230 ;    WRITE ("BEZEICHNUNG  : "); READLN (A↑.ARTIKEL);
240 ;    WRITE ("PREIS        : "); READLN (A↑.PREIS);
250 ;    A↑.NAECHSTES:= START; START:= A;
260 END;
270 IF MODUS="B" THEN BEGIN
280 ;    WRITE ("ARTIKELBEZEICHNUNG: "); READLN (BEZEICHNUNG);
290 ;    A:= START;
300 ;    WHILE A<>NIL DO
310 ;        BEGIN
320 ;        IF BEZEICHNUNG=A↑.ARTIKEL THEN
330 ;            BEGIN WRITELN ("ART.NR.: ",A↑.NR);
340 ;            WRITELN ("PREIS  : ",A↑.PREIS);
350 ;            WRITELN ("WEITER?");
360 ;            WHILE WEITER<>"J" DO BEGIN GET WEITER; END;
370 ;        WEITER:= "N"; END; A:= A↑.NAECHSTES;
380 ;        END;
390 ;        END;
400 IF MODUS="C" THEN BEGIN
410 ;    WRITE ("ARTIKELNUMMER: "); READLN (GESUCHT);
420 A:= START;
430 ;    WHILE A<>NIL DO
440 ;        BEGIN
450 ;        IF GESUCHT=A↑.NR THEN
460 ;            BEGIN WRITELN ("ARTIKEL: ",A↑.ARTIKEL);
470 ;            WRITELN ("PREIS  : ",A↑.PREIS);
480 ;            WRITELN ("WEITER?");
490 ;            WHILE WEITER<>"J" DO BEGIN GET WEITER; END;
500 ;        WEITER:= "N"; END; A:= A↑.NAECHSTES;
510 ;        END;
520 ;        END;
530 END;
540 END.
```

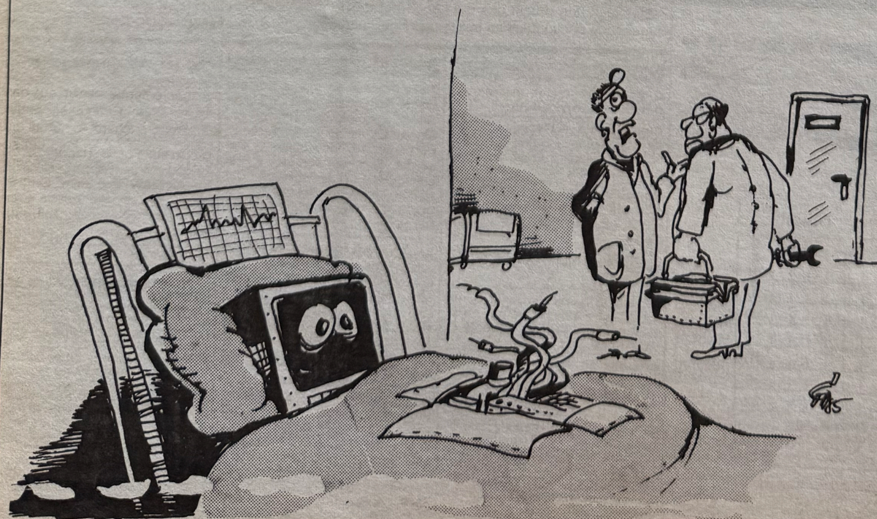
schiedliche BASIC-Dialekte, so ist ein gewisses Maß an Umdenken doch unerläßlich, wenn man zwischen zwei Computern wechselt, die beide Pascal „sprechen". Beispiel: Bei vielen Compilern kann man in Listen Speicherraum, den man nicht mehr benötigt, wieder freigeben. Dazu gibt es den Befehl DISPOSE. „Pascal 64" kennt diesen Befehl nicht. Einmal festgelegter Speicherraum bleibt festgelegt. Die einzige Ausweglösung ist, alle freigewordenen Listenelemente in einer weiteren Liste zusammenzufassen und von hier aus wieder zuzuweisen. Das ist recht umständlich, und man wird von dieser Möglichkeit wohl nur Gebrauch machen, wenn die Listen so umfangreich sind, daß man an akutem oder chronischem Speicherplatzmangel leidet.

Noch eine Aufgabe zur Übung:

Bisher wurde das neue Element immer am Anfang der Liste eingefügt. Schreiben Sie doch einmal ein Programm, das aus drei Teilen besteht:

a) Eingeben der Listenelemente
b) Einfügen weiterer Elemente an einer gewünschten Stelle
c) Ausdrucken der Liste, der Reihenfolge nach

Viel Erfolg! ■



Einführung in Forth

Teil 4

Logische Entscheidungen II. In der letzten Folge haben wir einfachste Vergleiche mit den Rechenoperationen = kennengelernt. Jetzt wollen wir verschachtelte IF...THEN Ausdrücke einmal in einem kleinen Programm anwenden. Das bedeutet, trifft eine Bedingung nicht zu, so sollen andere Abfragen erfolgen. Das nachfolgende Programm soll feststellen, welche Zahl welchem Monat zugeordnet ist. Wir sagen doch bekanntlich für den 1. Dezember auch 1. 12. Folglich steht 12 für Dezember, 11 für November und so weiter.

Von Uwe Haferland

```
DUP 12 = IF ." DEZEMBER"   ELSE
DUP 11 = IF ." NOVEMBER"   ELSE
DUP 10 = IF ." OKTOBER"    ELSE
DUP  9 = IF ." SEPTEMBER"  ELSE
DUP  8 = IF ." AUGUST"     ELSE
DUP  7 = IF ." JULI"       ELSE
DUP  6 = IF ." JUNI"       ELSE
DUP  5 = IF ." MAI"        ELSE
DUP  4 = IF ." APRIL"      ELSE
DUP  3 = IF ." MAERZ"      ELSE
DUP  2 = IF ." FEBRUAR"    ELSE
     1 = IF ." JANUAR"
THEN THEN THEN THEN THEN THEN
THEN THEN THEN THEN THEN THEN ;
```

Gestartet wird das Programm mit einer Monatszahl, gefolgt von einer Leertaste und dem Definitionswort MONAT. Betätigen Sie bitte nach jeder Zeile die ENTER- oder RETURN Taste. Wenn ich von Zeilen- oder Zeilennummern bisher gesprochen habe, so meine ich Zeilen auf Ihrer Seite. Die Zahlen am Anfang einer Zeile im Grafikprogramm der letzten Ausgabe dienten nur als Bezug und sollten daher nicht abgetippt werden. Haben Sie es trotzdem getan, so war dieses zufällig ohne Folgen. Wir befinden uns nämlich noch nicht im Programmodus, der uns erlauben würde, ein geschriebenes Programm, was sich in einem Block (oder mehreren) befindet, abzuspeichern.

An dieser Stelle seien noch einige Bemerkungen zum Editor gemacht. Ein professionelles FORTH - System kann je nach Speicherplatz bis zu 500 Blöcke enthalten. Jeder Block besteht aus 16 Zeilen (0 - 15), die wiederum je 64 Zeichen aufnehmen können. Folglich umfaßt ein Block 16 * 64 Byte = 1024 Byte [illegible] LIST listet den mit [illegible] angegebenen Block [illegible] lennummern 0 - 15 dann auf.

Ebenfalls sei darauf hingewiesen, daß der Forth Standard keine bestimmten Editor - Befehle vorschreibt! Hier entnehme man die entsprechenden Befehle der Bedienungsanleitung zur Forth-implementation. Wegen diesem heillosen Durcheinander werde ich Editorbefehle kaum behandeln. Nun wieder zurück zu unserem Programm. Die ELSE - Anweisungen veranlassen, daß bei einer nicht zutreffenden Bedingung ansonsten der nächste Befehl, also in unserem Fall der nächste Vergleich, stattfindet. Nach der letzten Bedingung 1 = IF. „Januar" darf kein ELSE - Befehl stehen, da THEN Kommandos folgen, genauer gesagt, zwölf Stück. Es müssen am Schluß solcher verschachtelten Bedingungen immer so viele THEN stehen, wie IF - Befehle verwendet wurden. Bekanntlich gehören IF..THEN immer zusammen.

Wo Kopieren noch erlaubt ist

Vor den Bedingungen wird immer die Kopieranweisung DUP verwendet, weil ein Vergleich die zu prüfende Größe zerstört. Dieser Prüfling wird vom Stapel genommen und nach dem Test, egal ob dieser positiv oder negativ verlief, vernichtet. Sollen noch weitere Vergleiche erfolgen, müssen wir uns Sicherheitskopien zulegen. Aber halt, haben wir denn vor dem letzten Vergleich nicht DUP vergessen? Nein, elf Kopien plus dem [illegible] Prüfling reichen für zwölf Vergleiche [illegible] wenn bei dem letzten die auszuwertende Zahl verschwindet. Dies ist uns aber egal, da keine

Korrektur zu Forth Teil 2

Leider haben sich bei der Drucklegung einige Fehler eingeschlichen, die an dieser Stelle berichtigt werden wollen.

```
Zu 1)
Berechnung von 3*4+2 mit Ausdruck des Zwischenergebnisses:
Loesung: 2 3 4 * DUP . + .
Die Stapelstruktur im Verlauf:

Am Anfang,nach * Operation,nach DUP,nach dem Punkt,nach +,
4            12                12         12          14
3             2                12          2
2                               2

nach dem letzten Punkt ist der Stapel leer.

2)
Anfangsstapelstruktur,nach ROT,nach SWAP,nach /,nach +, nach
3                   9         3        3      10      11
7                   3         9        7      11
9                   7         7       11
11                 11        11
```

Korrektur zu Forth Teil 3

Der Ausgabebefehl muß allgemein folgende Form haben: ." beliebiger Text"

Nach dem ersten Anführungszeichen muß unbedingt ein Leerzeichen folgen. Bitte bedenken Sie auch, daß nach jedem Befehl mindestens eine Leertaste betätigt werden muß. Auch vor dem Semikolon am Ende einer Definition muß sich ein Leerzeichen befinden! Der Punkt und das erste Anführungszeichen des Ausgabebefehls werden allerdings nicht durch ein Leerzeichen getrennt! In Zukunft werden die FORTH-Beispiele als Screen - Foto abgebildet, sodaß der Fehlerteufel keine Chance hat. Soweit die Berichtigung. Wir bitten um Verständnis.

weiteren Tests mehr erfolgen. Die Sache hat sogar noch den Vorteil, daß wir einen geräumten Stapel hinterlassen. Man sollte sich allgemein bei längeren Programmen angewöhnen, den Stapel, wo natürlich möglich, immer von nutzlosen Zahlen zu befreien, da der Stack nur eine begrenzte Anzahl an Elementen aufnehmen kann. Und wenn man diesen als Müllkippe benutzt, braucht man sich über einen Programmabbruch mit Fehlermeldung nicht zu wundern. Soweit die Vergleiche.



Für solche Ausdrucke ist Forth nicht besonders geeignet. Zu sehen war die Computerkunst auf der Internationalen Computer Show in Köln

Bekanntschaft mit dem Returnstack

An dieser Stelle lernen wir einen weiteren Stapel kennen, den Returnstack. Er enthält die ganzen Sprungadressen, die das Forthsystem bei der Abarbeitung der Befehle benötigt. Er steuert auch Schleifen, in denen er ihnen mitteilt, ob sie abgebrochen werden sollen oder nicht. Dieser Stapel ist wie der Parameterstack Lifo - orientiert. Wenn man auf dem Returnstack Zahlen abspeichern will, muß man hierbei einige Regeln beachten. Die gespeicherten Daten müssen noch vor Ende einer Definition wieder von ihm entfernt werden, denn das Forthsystem erwartet dort eine Rücksprungadresse. Wurde diese nun durch Ablage von Zahlen verändert, so geistert der Mikroprozessor ziellos durch alle Speicherzellen. Die Folge ist ein Systemsturz! Um diesen Stapel zu benutzen, stehen folgende Anweisungen zur Verfügung:

>R Das oberste Element des Parameterstacks wird auf den Returnstack abgelegt.
R> Die oberste Zahl des Returnstacks wird auf den anderen Stapel gelegt.
I Die oberste Date des Returnstacks wird kopiert und auf dem Parameterstack abgelegt.
I' Die zweitoberste Zahl des Returnstacks wird kopiert und auf dem Parameterstapel abgelegt.
J Die ganze Sache wie bei I oder I', allerdings nur mit dem drittobersten Element des Returnstacks.

Schauen wir uns jetzt einmal eine festgelegte Schleife an. Festgelegt bedeutet, daß schon am Anfang des Programms feststeht, wie oft die Schleife durchlaufen wird. Dabei wollen wir die Vorgänge auf dem Returnstack genauer untersuchen:

```
: TEST 20 0 DO I . LOOP ;
```

DO LOOP sind die Schleifenbefehle. Was dazwischen steht, sind die mehrfach zu wiederholenden Anweisungen. Die Zahl 20 in dem oben aufgeführten Beispiel stellt die Grenze dar, bei der abgebrochen werden soll, die 0 wird als Index oder auch als zu erhöhende Zahl bezeichnet. Der Befehl I macht uns eine Kopie vom obersten Element des Returnstacks, genauer gesagt vom Index, und legt diese auf den Parameterstack. Damit wir diese Zahl auch zu Gesicht bekommen, darf der Ausgabebefehl ''.'' nicht fehlen. Wir starten das Programm duch Eingabe des Definitionsnamens TEST. Sofort werden alle Zahlen von null bis neunzehn auf dem Bildschirm ausgegeben. Viele Leser werden Nummern von null bis zwanzig erwartet haben. Dies ist aber nicht der Fall, denn sobald bei Erreichen der LOOP Anweisung der Index von neunzehn auf zwanzig erhöht wurde, ist die Grenze erreicht, und der nächste Befehl nach LOOP wird abgearbeitet. Die Kommandos I . werden folglich gar nicht mehr erreicht. Soll Ihr Programm aus irgendeinem Grund bis zwanzig zählen, so hätten Sie als Grenze 21 eingeben müssen. Merken Sie sich auch bitte, solange nur Index und Grenze eingegeben werden, wird immer in Einerschritten aufwärts gezählt. Nutzen wir nun einmal die Schleife mitsamt der Kopie aus, um uns den internationalen Ascii - Code auszudrucken. Diesen Code kann man immer gebrauchen, insbesondere bei Grafikprogrammen (siehe HCR Ausgabe Nr. 7)

```
: ASCII 128 33 DO CR I DUP .'' CODE '' . ''.'' = '' EMIT LOOP ;
```

Wer nicht immer erst die Forthimplementation laden will, um ein paar ASCII - Codes zu erfragen, nehme nachfolgendes Basicprogramm:

```
10 FOR A = 33 TO 127
20 PRINT ''CODE'';A;'' = '';CHR$(A)
30 NEXT A
```

In beiden Fällen werden wir Tabellen auf dem Bildschirm oder Monitor erhalten, die wie folgt aussehen, wobei aus Platzgründen nur ein Ausschnitt aufgeführt sei. Bekanntlich ist der Ascii-Code ein 7-Bit Code, der nur Werte von 0 - 127 annehmen kann. In den Programmen habe ich aber erst die Codes ab 33 aufgeführt, da darunter die wenigen interessanten Steuerzeichen liegen, die je nach Computermodell und Implementation einen Systemabsturz verursachen können. Stellen Sie ein paar Experimente an, indem sie als Index immer kleinere Zahlen (bis null) einsetzen.

```
CODE 33 = !
CODE 34 = "
CODE 35 = #
CODE 36 = $
CODE 37 = %
CODE 38 = &
CODE 39 = ,
CODE 40 = (
```

Wer in dem Forthbeispiel nicht immer die Werte 128 und 33 verbessern will, um einen anderen Zahlenbereich zu erfassen, der gibt am besten das leicht modifizierte Programm ein:

```
: ASCII DO CR I DUP .'' CODE '' . '' '' = '' EMIT LOOP ;
```

Allerdings muß man den Wertebereich noch vor den Definitionsnamen eingeben, so daß ein Programmstart folgendermaßen aussieht: 128 20 ASCII.

Nun wird der Ascii - Bereich von 20 bis 127 angezeigt, vorausgesetzt, es erfolgt kein Systemabsturz. Doch wie läuft das Programm ab? Die beiden Zahlen 128 und 20 liegen erst einmal auf dem Parameterstack. Taucht eine Schleife auf, so meldet sich der Returnstack und nimmt sich die beiden obersten Zahlen des anderen Stapels. Mit diesen beiden Werten kann nun die Schleife beginnen. Solange die Grenze 128 nicht erreicht ist, werden alle Befehle zwischen DO und LOOP abgearbeitet. Im einzelnen bedeutet dies: Ein Zeilenvorschub erfolgt erst einmal. Dann wird der oberste Wert des Returnstacks kopiert (mit I), also der sich immer erhöhende Index, und auf dem Parameterstapel abgelegt. Da wir aber diese Zahl zweimal benötigen, nämlich für Ausdruck auf dem Bildschirm und für den EMIT - Befehl, müssen wir diese kopieren (DUP). Statt DUP hätten wir in diesem Fall auch I nehmen können. Nach der Vervielfältigung des Indexes wird das Wort ,,CODE'' mit Leerzeichen ausgedruckt. Dann folgt die Ausgabe des Indexes mit Hilfe des Befehls '',,''. Anschließend erscheint noch ein Gleichheitszeichen auf dem Bildschirm. EMIT nimmt sich den noch übrig gebliebenen Index vom Parameterstack und druckt das dazugehörige Ascii - Zeichen aus.



Für den Schneider gibt es derzeit noch nicht die Sprache Forth

LOOP erhöht den Index um eins, man spricht bei einer solchen Erhöhung auch von Inkrementieren, und wenn die Grenze noch nicht erzielt wurde, werden alle Kommandos nach DO noch einmal abgearbeitet. Dieses Spiel wiederholt sich solange, bis die Grenze erreicht ist. Da nach LOOP keine Befehle mehr stehen, würde in diesem Fall das Programm von selbst stoppen. Man sollte sich unbedingt merken, daß eine Schleife immer mindestens einmal durchlaufen wird, da der Vergleich Index = Grenze? erst bei LOOP erfolgt.

Veränderung der Schrittweite

Angenommen, der Index soll sich aus irgendeinem Grund nicht in Einerschritten verändern. Dazu dient die Anweisung +LOOP anstatt LOOP. Ferner sollen alle Parameter, also Index, Grenze und Schrittweite (= Inkrement) immer verändert werden. Dazu empfiehlt es sich, die Parameter vor dem Definitionsnamen einzugeben.

Entwerfen wir zuerst aber dazu das kleine Programm:

```
: TEST DO I . DUP +LOOP DROP ;
```

1.) Es soll in Zehnerschritten von -100 bis 100 gezählt werden.

Lösung:

10 110 - 100 TEST

2.) Es soll abwärts von 200 bis -200 in Viererschritten gezählt werden. Lösung: -4 -204 200 TEST

Obwohl nur bis -200 abwärts gezählt werden sollte, mußte die Grenze mit -204 festgesetzt werden, damit die -200 noch angezeigt wird. Merken Sie sich bitte auch bei der Eingabe die Reihenfolge: Inkrement - Grenze - Index. Den Index kann man auch als Anfangszahl bezeichnen, die bei jedem Schleifendurchgang verändert wird. Ihnen wird sicher der Kopierbefehl DUP aufgefallen sein. Das hat folgenden Grund: + LOOP erwartet ein Inkrement. Folglich wird es vom Parameterstapel genommen und dadurch vernichtet. Damit bei dem nächsten +LOOP noch die Schrittweite vorhanden ist, muß diese vor der nächsten +LOOP Anweisung kopiert werden. Beachten Sie bitte folgende Vorgänge: Das Inkrement liegt immer unter der Grenze und dem Index. Ein LOOP - Befehl nimmt diese beiden oberen Elemente und legt sie auf den Returnstack. Nun befindet sich das Inkrement ganz alleine auf dem Stapel. Zwar sorgt der I - Befehl für kurzzeitigen Besuch, indem eine Kopie des Indexes auf dem Parameterstack abgelegt wird, jedoch verschwindet diese Date nach dem Ausgabebefehl. Da das Inkrement jetzt wieder alleine auf dem Parameterstapel liegt, und +LOOP sich nähert, muß es jetzt kopiert werden. Da nach dem letzten +LOOP die Kopie auf dem Stack nicht mehr ausgegeben werden kann, sollte sie noch vor Beendigung des Programms mit DROP vom Stack entfernt werden, damit bei mehrmaligen Programmstarts der Parameterstapel nicht zur Müllkippe wird.

Endlosschleifen

Für einige Zwecke ist es sinnvoll, eine Schleife unendlich lange laufen zu lassen. Dazu nimmt man als Schrittweite einfach null. Der Grund für dieses Endlosverhalten liegt in dem Inkrement 0, das den Index in Nullerschritten erhöht, also gar nicht. Das so nie eine Grenze erreicht werden kann, versteht sich von selbst. Abschließend sollte man sich noch merken, daß als Grenzen für eine Schleife Zahlen von -32768 bis 32767 in Frage kommen. Soll eine Schleife zum Beispiel 100000 mal durchlaufen werden, so reicht dieser Zahlenbereich nicht aus und man verwendet wie in Maschinensprache verschachtelte Schleifen, die wir in der nächsten Folge uns näher betrachten wollen, zusammen mit nicht festgelegten Wiederholungsanweisungen (BEGIN UNTIL und WHILE REPEAT). Ebenfalls werden wir noch Tricks kennenlernen, wie man eine Endlosschleife doch noch verlassen kann.

Fortsetzung folgt. ■

Kraftwerke

Obwohl Heimcomputer in den letzten Jahren immer komfortabler und leistungsfähiger geworden sind, repräsentieren die dazu gehörigen Netzteile in der Regel den Stand der Technik von vor fünfzehn Jahren. In seltenen Fällen nur sind die Netzteile im Rechnergehäuse untergebracht, und dies nicht nur aus Platzgründen. Selbst Hersteller wie Commodore, die ja wirklich viel umbaute Luft in ihren Computergehäusen haben, liefern externe Netzteile. Beim Sinclair ZX Spectrum gar ist das Netzteil nichts anderes als das, was wirklich unbedingt notwendig ist, um aus 220 Volt Wechselspannung eine für den Rechner verträgliche Gleichspannung zu machen. Dies kann, besonders bei Microdrivebetrieb, zu Schwierigkeiten führen

Von T. Schwarz

Wie ist nun ein richtiges Computernetzteil aufgebaut?

Sicherlich hat schon der eine oder andere einmal den Begriff «Schaltnetzteil» gehört. Nun will ich hier nicht die Schaltungstechnik dieser Netzteile erläutern. Da diese nur als Beispiel dienen, ist wichtig, was hinten herauskommt. Wichtig sind aber auch noch die wesentlichen Kriterien, die ein solches Schaltnetzteil erfüllen sollte:

Konstante Ausgangsspannung

Eigentlich eine Selbstverständlichkeit, daß ein Computernetzteil keine Spannungen erzeugen sollte, die nur grob geschätzt werden können, sondern präzise, geregelte, in minimalen Toleranzen gleichbleibende Spannungen.

Strombegrenzung

Um Schäden am Netzteil selbst, sowie auch um Schäden am Rechner zu verhindern, muß ein Netzteil natürlich unbedingt irgendeine Art Strombegrenzung haben. Dies kann eine elektronische Schutzschaltung sein, die einfachste und bekannteste Form der Strombegrenzung ist immer noch eine Sicherung.

Dies sind nur die wichtigsten Kriterien. Schaltnetzteile mit sogenannten Durchflußwandlern benötigen auch noch einen SOFT-START, das heißt, die Leistung wird langsam hochgefahren. Nötig ist dies bei den Durchflußwandlern, weil deren ungeladene Filterkondensatoren beim Einschalten praktisch wie ein Kurzschluß wirken; aber auch bei anderen Bauformen von Computernetzteilen ist ein SOFT-START angenehm.

Weiterhin sollte ein Netzteil natürlich auch noch über einen Ein/Aus-Schalter verfügen, damit man nicht den Stecker ziehen muß, um es abzuschalten.

Nun seien Sie einmal ehrlich, und untersuchen Sie einmal das Netzteil IHRES Heimcomputers:

-Ist es im Rechnergehäuse untergebracht oder ist es separat?
-Hat es einen Ein/Aus-Schalter?
-Hat es eine Kontrolleuchte?
-Erzeugt es eine konstante Spannung?
-Besitzt es eine Strombegrenzung?

Ich bin sicher, daß die meisten Netzteile für Heimcomputer nicht alle diese Kriterien erfüllen.

Ich habe nun das Netzteil des Sinclair ZX Spectrum einmal geöffnet, hineingesehen und gemessen. Ich war auf der Suche nach der Ursache für eine Störung des Microdrives. Ist nämlich beim Ein- oder Ausschalten des Spectrums eine Cassette in den Drive eingelegt, können *hierauf gespeicherte Daten verlorengehen oder verstümmelt werden. Nicht immer, aber einmal ist ja schon zuviel.*

Das Spectrum-Netzteil besteht aus den Grundbausteinen eines Netzteiles: einem Transformator, vier Dioden und einem Kondensator. In einem Netzteil geschieht nämlich folgendes: Die angelegte Netzspannung, in unserem Fall 220 Volt, wird in eine Wechselspannung geringerer Größe umgewandelt, im Falle des Spectrum 9 Volt. Dieses besorgt der Transformator. In den Ausgang des Transformators werden Dioden geschaltet, um aus der Wechselspannung die für den Computerbetrieb erforderliche Gleichspannung zu machen. Es muß mindestens eine Diode sein; allerdings ist eine solchermaßen erzeugte Gleichspannung für den Computer nicht zu gebrauchen. Also macht man mit vier Dioden eine sogenannte Brückenschaltung. Auch Gleichrichter sind vier Dioden in Brückenschaltung, nur in einem Gehäuse. Nachdem die Spannung gleichgerichtet ist, besitzt sie noch gewisse Oberwellen, denn sie war ja ursprünglich eine sinusförmige Wechselspannung. Also glättet man diese Wellen mit einem Kondensator, dem Glättungskondensator.

Die Grundschaltung

Nun, dies ist die Grundschaltung eines Netzteils, und dies ist auch die Schaltung des Spectrum-Netzteiles. Auf dem Typenschild dieses Netzteiles sind jetzt 9 Volt als Ausgangsspannung angegeben; weil der Glättungskondensator aber großzügig dimensioniert ist, bewirkt er eine Spannungsüberhöhung, so daß die Ausgangsspannung gelegentlich auch schon einmal 15 Volt betragen kann. Dies ist keine Spekulation, diese Werte habe ich an verschiedenen Netzteilen gemessen. Diese Spannung ist überdies auch noch lastabhängig, also je größer die Stromaufnahme des Computers ist (sprich: je voller der Speicher), desto mehr sinkt die Spannung ab.

Wie vielen sicherlich bekannt ist, beträgt die Versorgungsspannung für IC's, wie sie ja auch im Spectrum Verwendung finden, 5 Volt. Da aber nun 15 Volt anliegen, muß mit den überschüssigen 10 Volt ja irgendetwas geschehen.



Unser Bild zeigt eine problematische Anwendung für das Spectrum - Netzteil

10 Volt zuviel

Und es geschieht auch etwas. Die Spannungsregulierung erfolgt nicht, wie man eigentlich vermuten sollte, intern im Netzteil, sondern im Rechner selbst. In der rechten unteren Ecke, etwa in Höhe des Beepers, sitzt ein auf ein Kühlblech montierter integrierter Spannungsregler, auch Festspannungsregler genannt. Dieser bewirkt nun, daß unabhängig von der Eingangsspannung die Ausgangsspannung immer konstant 5 Volt beträgt. Es muß aber die unerwünschte überschüssige Differenzspannung irgendwie verschwinden. Nun, diese wird regelrecht verbraten, sie wird nämlich in Wärme umgewandelt. Dies hat unter anderem zur Folge, daß das Spectrum-Gehäuse in der rechten unteren Ecke nach längerem Betrieb bedrohlich warm wird und sich die Gummifüße des Gehäuses langsam ablösen. Auch die so schon wabbelige Tastatur verwandelt sich in der rechten unteren Ecke vom Radier- langsam in ein Kaugummi.

Der Umbau

Wer sich nun einmal mit dem Schaltplan des Spectrums beschäftigt hat, der wird festgestellt haben, daß zum Beispiel am Video-Chip teilweise eine Spannung von 12 Volt anliegt. Diese Spannung wird jedoch intern im Rechner erzeugt; um sicher arbeiten zu können, benötigt der Spectrum tatsächlich nur eine Spannung von 5 Volt. Was liegt also nun näher, als das Netzteil des Spectrums derart umzubauen, daß es auch tatsächlich als Netzteil bezeichnet werden kann, zumal, wenn man sich die Nachteile vor Augen führt:

-Datenzerstörung auf dem Microdriveband
-starke Wärmeentwicklung im Innern des Computers, wodurch sich die Standfüße ablösen und auch die Tastatur leidet

- keine Sicherung
- kein Ausschalter
- keine Kontrollleuchte

Besonders die fehlende Sicherung ist ein wirklich schwerwiegender Mangel, der schon des öfteren zur Zerstörung des ROM geführt hat. Einmal eine Erweiterung falsch angesteckt, einmal ein Fehler am Festspannungsregler, und schon hat der Spectrum nur noch Schrottwert. ■

Basic-Rätsel

Da der PRINT—Befehl wohl einer der wichtigsten Befehle in unserer Basic-Programiersprache ist, wollen wir den Schwerpunkt auch in diesem Basic-Rätsel darauf legen. Wir suchen 95 Werte des SINUS, das ist ein ganzer Bildschirm voll (25 Zeilen/40 Zeichen).

Es sollen also jeweils die Schritte und daneben der fünfstellige Sinuswert ausgegeben werden. Stehen alle 95 Positionen auf dem Bildschirm, soll erst nach einem Tastendruck die READY-Meldung erscheinen. Hier ein Beispiel der ersten drei oberen Bildschirmzeilen:

1	.0174	25	.4363	49	.8552	73	1.274
2	.0349	26	.4537	50	.8726	74	1.291
3	.0523	27	.4712	51	.8901	75	1.308

Als Hilfe, die Sinus-Werte errechnen wir "PI/180*1 bis 95"

Lösung für Schneider CPC 464

```
10 INPUT "ZUERST DEN KM-STAND VOM ERSTEN TANKEN EINGEBEN, DANN VOM ZWEITEN, DANN IE GE- TANKTEN LITER UND DANN DEN DM-BETRAG DER LITER EINGEBEN (Z.B. 74500,74950,45,63)";a,b,L,DM
20 X = B - A : L * 100 / X : Z = DM / X
30 PRINT : PRINT" DER WAGEN VERBRAUCHTE AUF 100 KM ", Y;"LITER, PRO KM SIND DAS ";Z;"DM."
```

Annette Kasper

Lösung für Schneider CPC 464

```
10 CLS : INPUT "KM-STAND 1. TANKEN "; K1 : PRINT : INPUT "KM-STAND 2. TANKEN "; K2 : PRINT : INPUT "LITER ";l : PRINT : INPUT "TANKBETRAG ";t : PRINT : PRINT "VERBRAUCH/100 KM ="; (l * 100)/(K2-K1); " LTR." : PRINT : PRINT "BENZINKOSTEN/KM =";t/(K2-K1);"DM"
```

Olaf Herden

Lösung für Commodore 64

```
10 INPUT"(CLR SCREEN) KM-STAND BEIM 1. TANKEN:";A: INPUT"KM-STAND BEIM 2. TANKEN:";B
20 INPUT"GESAMTLITERZAHL:";C: INPUT"TANKBETRAG:";D: E = B - A : F = E / C : G = D / C : H = G / F
30 PRINT"VERBRAUCH / 100 KM:";F;"LITER" : PRINT"BENZINKOSTEN PRO KM:";H;"DM"
```

Alexander Kraemer

Lösung für Schneider CPC 464

```
1 INPUT"(CLR SCREEN)KM1,KM2,LTR,DM";A,B,C,D : E = B - A : PRINT"(CRSR DOWN) = "100 * C / E"LTR/100 KM = "D / E "DM/KM
```

Willi Boosz

Lösung für IBM PC

```
10 INPUT "km-Stand 1. Tanken   : ",KM1:INPUT "km-Stand 2. Tanken   : ",KM2:INPUT "Liter               : ",L:INPUT "Tankbetrag           : ",B:KM=KM2-KM1:PRINT "Verbrauch/100 km    =";L*100/KM;"ltr.":PRINT "Benzinkosten pro km =";B/KM;"DM"
```

Horst May

Lösung für Commodore 64

```
1 INPUT"(CLR SCREEN)(CRSR DOWN)KM-STAND 1. TANKEN:";T1 : INPUT"KM-STAND 2. TANKEN:";T2 : PRINT"? LITER:";
2 INPUTTL : INPUT"TANKBETRAG:";TB : PRINT"(CRSR DOWN)(CRSR DOWN)VERBRAUCH / 100 KM:" ;: V = (TL / (T2-T1)) * 100
3 PRINTINT(V * 100 + .5) / 100 : PRINT"BENZINKOSTEN PRO KM:"INT ((TB / TL) * 100 + .5) /100"DM"
```

Gerd Bleckwedel

Lösung für TI 99/4A

```
10 INPUT "km-Stand:1.Tanken, 2.Tanken,
Liter, Betrag in DM ? ":KM1,KM2,L,B
20 PRINT ::"Verbrauch :";INT(L*100/(KM2
-KM1, )+.5)/10;"l/100 km"::"Benzinkos
ten pro km";INT(B*100/(KM2-KM1)*10+.5)
/10;"Pf"::
```

Alfred Hein

Lösung für Schneider CPC 464

```
10 MODE 1:PRINT"km-Stand  1. Tanken :":PRINT"km-Stand  2.
   Tanken :":PRINT"Liter  :":PRINT"Betrag :":LOCATE 23,1:
   INPUT "",t1:LOCATE 23,2:INPUT "",t2:LOCATE 23,3:INPUT
   "",l:LOCATE 23,4:INPUT "",p
20 km=t2-t1:v=l/km*100:bk=p/km
30 PRINT:PRINT USING"Verbrauch/100 km    = ##.# ltr";v:
   PRINT USING"Benzinkosten pro km = #.##  DM";bk:PRINT
```

Manfred Memmler

Lösung für Commodore 64

```
1o  print"♡":Input"km-stand 1";a:Input"km-stand 2";b:
    c=b-a:print"gefahrene km";c

2o  Input"? liter";d:Input"Preis";e:l=e/d:v=d*1oo/c:Print
    "Verbr./1oo km:";v:k=e/c

3o  Print"Kosten pro km=";k;"DM"
```

Klaus-Ulrich Hartmann

Physik per Computer

Teil 3: Der schiefe Wurf

Fortsetzung von Seite 5

```
810 G=9.80665
820 FOR I=1 TO 3
830 FOR J=1 TO 3
840 H(I,J)=(V(J)*SIN(RAD(PHI(I))))↑2/(2*
G)
850 T(I,J)=2*V(J)*SIN(RAD(PHI(I)))/G
860 W(I,J)=V(J)↑2*SIN(2*RAD(PHI(I)))/G
870 NEXT J,I
880 IF P1=2 THEN 1040
890 GOSUB 2090:REM MAXIMUMBERECHNUNG
900 FOR K=1 TO 3
910 FOR I=1 TO 3
920 DT=T(I,K)/100
930 FOR J=0 TO 100
940 X(J,I)=V(K)*J*DT*COS(RAD(PHI(I)))
950 Y(J,I)=V(K)*J*DT*SIN(RAD(PHI(I)))-G/
2*(DT*J)↑2
960 NEXT J
970 NEXT I
980 GOSUB 1060:REM PLOTPROGRAMM
990 FOR I=1 TO 3
1000 FOR J=0 TO 100
1010 X(J,I)=0:Y(J,I)=0
1020 NEXT J,I
1030 NEXT K
1040 GOSUB 1670:REM DISPLAY/DRUCKPROG.
1050 END
1060 REM******************************
1070 REM
1080 REM     Subroutine PLOT
1090 REM     ---------------
1100 REM******************************
1110 REM
1120 PCOLOR 0
1130 MODE TS
1140 SKIP10
1150 IF K>1 THEN 1230
1160 MODE TN
1170 PRINT/PTAB(5)"S c h i e f e r  W u
r f"
1180 PRINT/PTAB(5)"================: =====
==="
1190 PRINT/P
1200 PRINT/PTAB(5)"1. Wurfparabeln"
1210 PRINT/PTAB(5)"---------------":PRIN
T/P
1220 MODE TS
1230 PRINT/PTAB(10)"Abschu";CHR$(174);"g
eschw. ";V(K);" m/s":PRINT/P
1240 FOR I=1 TO 3
1250 PCOLOR I
1260 PRINT/PTAB(10)"PHI(";I;") =";PHI(I)
;"Grd"
1270 NEXT I
1280 PRINT/P
1290 MODE GR
1300 PCOLOR 0
1310 XB=400/WM
1320 MOVE 50,-(HM*XB+20)
1330 HSET
1340 AXIS 1,80,5
1350 PHOME
1360 AXIS 0,(HM*XB/5),5
1370 PHOME
1380 MOVE-5,(HM*XB+15)"
1390 GPRINT"H/m"
1400 MOVE400,5
1410 GPRINT"W/m"
1420 PHOME
1430 LW=80
1440 LH=HM*XB/5
1450 FOR Z=1 TO 5
1460 MOVE (LW*Z-10),-15
1470 X1=WM/5*Z
1480 X1=INT(X1*100)/100
1490 X1$=STR$(X1)
1500 GPRINTX1$
1510 NEXT Z
1520 FOR Z=1 TO 5
1530 Y1=HM/5*Z
1540 Y1=INT(Y1*10)/10
1550 Y1$=STR$(Y1)
1560 MOVE -30,(LH*Z-5)
1570 GPRINTY1$

1580 NEXT Z
1590 FOR I=1TO 3
1600 PCOLOR I
1610 PHOME
1620 FOR J=0 TO 100
1630 LINE (X(J,I)*XB),(Y(J,I)*XB)
1640 NEXT J
1650 NEXT I
1660 RETURN
1670 REM******************************
1680 REM
1690 REM   Subroutine DISPLAY/DRUCK
1700 REM   ------------------------
1710 REM******************************
1720 CLS
1730 PRINT:PRINT:PRINT
1740 IF P1=2 THEN 1780
1750 MODE TN
1760 SKIP5
1770 PCOLOR 0
1780 PRINTTAB(5)"Schiefer Wurf"
1790 IF P1=1 THENPRINT/PTAB(5)"2. Wertet
abellen"
1800 PRINTTAB(5)"-------------"
1810 IF P1=1 THENPRINT/PTAB(5)"---------
-------"
1820 FOR I=1 TO 3
1830 PRINTTAB(4)"Abschu";CHR$(174);"gesc
hw. =";:PRINTUSING"###.##";V(I);:PRINT"
m/s":PRINT
1840 PRINTTAB(12)"   W/m     H/m      T/s"
1850 PRINTTAB(13)"----------------------
":PRINT
1860 FOR J=1 TO 3
1870 PRINT"PHI = ";:PRINTUSING"##.#";PHI
(J);:PRINT"'";
1880 PRINTTAB(4)" ";:PRINTUSING"#####.#"
;W(J,I),H(J,I),T(J,I)
1890 NEXT J
1900 PRINT
1910 PRINTTAB(10)"Weiter ? Leertaste ↓"
1920 GET LL$
1930 IF LL$<>" " THEN 1920
1940 NEXT I
1950 IF P1=2 THEN RETURN
1960 FOR I=1 TO 3
1970 PCOLOR1
1980 PRINT/P"Abschu";CHR$(174);"geschw.
=";:PRINT/PUSING"###.##";V(I);:PRINT/P"
m/s":PRINT/P
1990 PCOLOR0
2000 PRINT/PTAB(14)"    W/m     H/m     T
/s"
2010 PRINT/PTAB(16)"--------------------
--":PRINT/P
2020 FOR J=1 TO 3
2030 PRINT/P"PHI = ";:PRINT/PUSING"##.#"
;PHI(J);:PRINT/P" Grd";
2040 PRINT/PTAB(4)" ";:PRINT/PUSING"####
#.#";W(J,I),H(J,I),T(J,I)
2050 NEXT J
2060 PRINT/P
2070 NEXT I
2080 RETURN
2090 REM*****************************
2100 REM

2110 REM    Subroutine MAXIMUM
2120 REM    ------------------
2130 REM*****************************
2140 REM
2150 HM=-1E+10
2160 WM=-1E+10
2170 FOR I=1 TO 3
2180 FOR J=1 TO 3
2190 IF H(I,J)>=HM THEN HM=H(I,J)
2200 IF W(I,J)>=WM THEN WM=W(I,J)
2210 NEXT J
2220 NEXT I
2230 RETURN
```

2. Wertetabellen

Abb.5.

Abschußgeschw. =130.00 m/s

	W/m	H/m	T/s
PHI = 30.0 Grd	883.1	127.5	10.2
PHI = 45.0 Grd	1019.7	254.9	14.4
PHI = 85.0 Grd	177.1	506.0	20.3

Abschußgeschw. =125.00 m/s

	W/m	H/m	T/s
PHI = 30.0 Grd	1379.8	199.2	12.7
PHI = 45.0 Grd	1593.3	398.3	18.0
PHI = 85.0 Grd	276.7	790.6	25.4

Abschußgeschw. =150.00 m/s

	W/m	H/m	T/s
PHI = 30.0 Grd	1987.0	286.8	15.3
PHI = 45.0 Grd	2294.4	573.6	21.6
PHI = 85.0 Grd	398.4	1138.5	30.5

1. Wurfparabeln

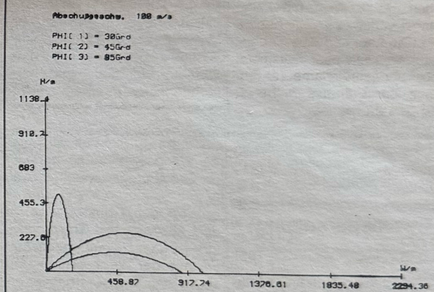


Abb.2.

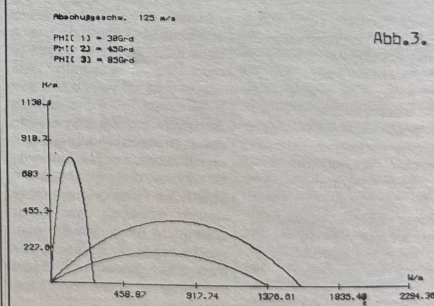


Abb.3.

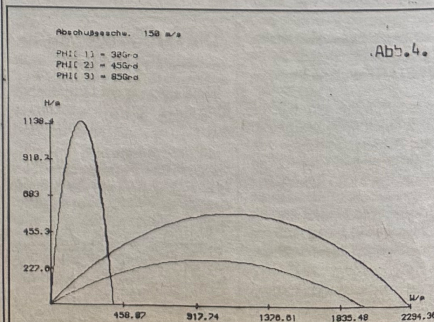


Abb.4.

—Die 68000 Dimension—

— Teil 3 — Coprozessormöglichkeiten

Den wahrscheinlich größten Vorteil bieten die 16 Bit CPUs durch die Erweiterungsmöglichkeit mit einem Coprozessor, wobei es sich genauer gesagt um einen Arithmetikchip handelt. Doch was bedeutet das? Dazu muß ich weiter ausholen und die Problematik bei der Verarbeitung von Gleitkomma Operationen schildern. Wenn man mit Integerzahlen rechnet, das sind Zahlen ohne Kommas, kann ein Mikroprozessor jeder Klasse diese mit relativ wenigen Befehlen verarbeiten.

Von Uwe Haferland

Dadurch erreicht man hohe Verarbeitungsgeschwindigkeiten. Das erkennt man an den Integercompilern, die gegenüber ihren Floating - Point (Komma) Kollegen wahre Geschwindigkeitsrekorde aufstellen. Unbequem wird es für die Prozessoren, wenn sie kommabehaftete Größen (Floating - Point) verarbeiten sollen. Betrachten wir uns einmal dazu eine normalerweise einfache Addition. Ein Summand (mindestens zwei existieren) besteht bei Gleitkommaverarbeitung aus zwei Größen, der Mantisse und dem Exponenten. Die Mantisse nimmt folgende Werte an: 0.1 Mant. 1. Der Exponent gibt dagegen die Position des Kommas an. Will man nun zwei kommabehaftete Zahlen addieren, funktioniert es nur mit Exponenten gleicher Größe. Also werden die Mantissen solange umgeformt, bis die Exponenten übereinstimmen. Erst dann kann die eigentliche Rechenoperation ausgeführt werden. Es ist doch anhand des Beispiels leicht einzusehen, daß hierzu eine Menge Zeit benötigt wird, was die Rechengeschwindigkeit stark reduziert. Um diesem Nachteil zu entgehen, wurden Arithmetikprozessoren entwickelt, die die verschiedensten mathematischen Operationen mit Gleitkomma in kürzester Zeit behandeln. Die nachfolgende Tabelle gibt hierzu ein paar Beispiele, wobei als gemeinsame Taktfrequenz der Fairness halber 8 MHz gewählt wurde:

	68000	Z 80	8087
Mantissen	32 Bit	32 Bit	52 Bit
Addition	0.06 ms	0.4 ms	0.009 ms
Subtraktion	0.06 ms	0.4 ms	0.011 ms
Multiplikation	0.06 ms	1.3 ms	0.017 ms
Division	0.11 ms	2.2 ms	0.024 ms
Quadratwurzel	0.65 ms	8.0 ms	0.023 ms
Sinus	1.40 ms	19 ms	————
Exponentialfunktion	1.30 ms	14.5 ms	0.063 ms

Obwohl der 8087 eine größere Mantisse bearbeiten mußte, ist dieser Arithmetikprozessor in dem Vergleichstest, trotzdem noch wesentlich schneller als seine Konkurrenten. Ein solches Rechengenie wird immer zusammen mit einem normalen Prozessor angewendet, die sich, abhängig von den Befehlen, die Arbeit aufteilen. Beide erreichen dann sagenhafte Rechengeschwindigkeiten, wie das nachfolgende Beispiel zeigt:

80 Bit Division mit Gleitkomma

Intel 8085 (2.5 MHz) 60 ms; Intel 8086 3.2 ms; Intel 8086 + 8087 zusammen (5 MHz) 0.039 ms.

Ein Arithmetikprozessor arbeitet mit sehr breiten Registern, die beim 8087 achtzig Bit Zahlen aufnehmen können! Ein Nachteil solcher Coprozessoren sei jedoch nicht verschwiegen. Zahlt man für einen 16 Bit Prozessor so um die hundert Mark, kommt man bei den Mathematikgenies in der Regel gerade mit 500 - 600 DM aus. Der in unseren Beispielen aufgeführte 8087 Arithmetikprozessor ist im übrigen für eine Zusammenarbeit mit dem Intel 8086 und 8088 ausgelegt. Selbstverständlich läßt sich auch der Motorola 68000 mit einem solchen Rechengenie ausstatten, der nicht 8087 heißt, sondern 68881 vom gleichen Hersteller, aber gegenwärtig noch nicht lieferbar ist. Er kann für die gesamte 68000 Familie verwendet werden, also auch für den demnächst erscheinenden 32 Bit Prozessor 68020. Dieser frühestens ab Sommer 1985 erhältliche Arithmetikprozessor 68881 stellt wirklich alles bisher dagewesene in den Schatten. So besitzt er neben den vier Grundrechenarten: Wurzel, arccos, arcsin, arctan, arctanh, cos, cosh, exp(x), log, In, log mit festlegbarer Basis, sin, sin und cos gleichzeitig berechnen, sinh, tan, tanh, Potenzen mit der Basis 10 und2. Hier nun einige Berechnungszeiten, die einmalig sind:

Aufgabe (64 Bit Zahlen)	Ausführungszeit in us
Addition und Subtraktion	4.6
Multiplikation	5.8
Division	7.7
sin	24.9
cos	24.9
sin und cos gleichzeitig	27.4
tan	29.1
arcsin	35.7
arcos	37.7
arctan	25.5

Dieser Arithmetikprozessor besitzt acht Register mit einer Datenbreite von achtzig Bit. Er kann mit einer maximalen Taktfrequenz von sage und schreibe 16,67 MHz betrieben werden. Da Haupt- und Coprozessor ein asynchrones System bilden dürfen, kann der 68881 mit knapp 17 MHz laufen, während der 68000 sich mit niedrigeren Taktfrequenzen abgibt. Nur so entsteht ein Maximum an Rechengeschwindigkeit und Unabhängigkeit. An dieser Stelle wird sich mancher Leser sagen, gut, dann statte ich einen Mikroprozessor der acht Bit Klasse mit einer Arithmetikeinheit aus, so daß durch die daraus resultierende höhere Rechengeschwindigkeit ein 16 Bit Prozessor weitgehend überflüssig wird. Das ist leider nicht richtig. Der Haupt- und Coprozessor müssen sich nämlich den Datenbus teilen. Da nun der Datentransfer auf dem Datenbus eines 8 Bit Prozessors sehr langsam abläuft, verursacht durch die nur vorhandenen acht Datenleitungen und der recht kleinen Taktfrequenz, wird dieser Bus des Hauptprozessors fast die ganze Zeit blockiert, und läßt den Arithmetikprozessor kaum zum Zuge kommen. 16 Bit CPUs mit ihren meist sechzehn Datenleitungen holen sich immer gleich zwei Speicherzelleninhalte auf einmal. Und bei einer hohen Taktfrequenz von etwa acht bis zwölf Megahertz geschieht dies auch noch sehr geschwind. Kein Wunder also, daß die 16 Bit Mikroprozessoren den Datenbus sehr schnell wieder freigeben können. Hinzu kommen Macrobefehle, die den Hauptprozessor lange beschäftigen und daher quasi nicht den Datenbus in diesem Zeitraum benötigen. Außerdem unterstützen bei einigen 16 Bit Prozessoren bestimmte Befehle die Arbeitsteilung, so zum Beispiel der Maximum Modus beim 8086 und 8088.

Der Motorola 68000 intern

An dieser Stelle soll bevorzugt nur der 68000 betrachtet werden, da die 8 Bit Klasse nichts gleichwertiges aufbietet, und die anderen 16 Bit Typen nur vereinzelt mithalten können. Die Motorola CPU besitzt auffallend viele Interruptvektoren (192), die eine Unterbrechung mit definierten Sprung hervorrufen, sobald irreguläre Bedingungen wie zum Beispiel eine Division durch Null, unzulässige Befehlscodes oder Überschreitung vorgegebener Wertegrenzen in Datenregistern auftreten. Solche Unterbrechungen werden auch als Traps bezeichnet. Bei der 8 - Bit Klasse können solche Interrupts nur durch aufwendige Software erreicht werden, die aber den Prozessor sehr verlangsamen. Erstmalig tritt auch eine zweite arithmetisch - logische Einheit für die Adressierung auf. Eine ALU führt bekanntlich alle Rechenoperationen aus. Bei den früheren Mikroprozessoren mußten Daten und Adressen ,,Schlange stehen'', um verarbeitet zu werden. Dieser Zustand ist aber dem Motorola 68000 völlig unbekannt, da die Adressen eine eigene Rechenzentrale besitzen. Dies führt selbstverständlich zu einer erhöhten Arbeitsgeschwindigkeit.

Kannten die 8 - Bit Mikroprozessoren nur einen Stapel (= reservierte Speicherplätze für besondere Daten) und nur einen Stapelzeiger, der auf eine Adresse des Stapels zeigt, liegen bei Motorola dagegen völlig andere Strukturen vor. Ein lifo orientierter Stapel (was zuletzt an Daten in den Stapel gelegt wurde, kommt auch bei Aufruf wieder zuerst heraus), und ein fifo orientierter Stapel (= Gegenteil von Lifo), auch Queue genannt, erlauben in Verbindung mit den beiden Stapelzeigern ein sehr elegantes Programmieren, was kürzere Programme erlaubt und somit kleinere Laufzeiten.

Im Gegensatz zur 8 Bit Generation sind Befehle (Link und Unlink) beim 68000 vorhanden, die jedem Compiler erlauben, schnelleren Maschinencode zu produzieren. Das bedeutet, wer mit höhreren Programmiersprachen wie Pascal, Fortran oder Forth arbeitet, also sogenannten Compilersprachen (Compiler = Name des Übersetzers), kann Dank der Unterstützung einiger Maschinenbefehle mit einer hohen Verarbeitungsgeschwindigkeit rechnen.

Kosten

Leider kann ein solches Maximum an Technik nicht gerade preiswert erworben werden. Kostet ein 8 - Bit Prozessor zwischen 10 und 15 DM, muß man für den 68000 einen Hunderter hinblättern. Auch das im Sommer erscheinende Rechenwunder namens 68881 wird schätzungsweise mindestens 600 DM kosten, was für einen hochwertigen Arithmetikprozessor ein völlig normaler Preis ist (der 8087 ist nicht unter 600 DM erhältlich).



Für solche Berechnungen benötigt ein Arithmetikprozessor nur noch einen Teil der Zeit des Hauptprozessors

Software-kompatibilität

Da der Motorola 68000 von seinen Hard- und Softwareeigenschaften eine völlige Neuentwicklung darstellt, kann die bisherige Software für andere Mikroprozessoren leider nicht benutzt werden. Es sprechen aber alle Anzeichen dafür, daß der 68000 sich in der 16 - Bit Klasse durchsetzen wird. So existieren schon Betriebssysteme wie Unix oder CPM/68k, die den gleichen Effekt haben werden wie CPM beim Z 80, nämlich ein großes Softwareangebot. Mittlerweile erscheinen aber von bekannten Firmen wie Apple, Commodore und Atari Rechner mit dem Motorola - Prozessor, was in Kürze auch ein größeres Softwareangebot hervorrufen wird. So hat Atari laut Firmenaussage allein in den Vereinigten Staaten etwa 120 Firmen beauftragt, für den 520 ST (hat den 68000) Programme zu entwickeln. Existiert erst einmal ein solches für einen bestimmten Prozessor, so ist es nicht schwer, dieses für einen anderen Rechner mit der gleichen Zentraleinheit umzuschreiben. Das erlebt man doch fast jede Woche. Erscheint ein Programm für den Sinclair Spectrum, so ist es wenige Wochen später auch schon für den Amstrad (Schneider) erhältlich. Weitere zahlreiche Beispiele ließen sich noch aufführen.

Aussichten

Der Motorola 68000, der inzwischen auch in Lizenz von Valvo gebaut wird, stellt nicht den neuesten Stand der Technik dar. So wird Motorola frühestens im Sommer 1985 einen echten 32 - Bit Prozessor namens 68020 auf den Markt bringen. Er besitzt schon 64 - Bit Register und einen Adreßbus, der Speicher in der Gigabytegröße adressieren kann und einen 32 - Bit Adreßbus besitzt. Der Befehlssatz wurde nochmals wesentlich erweitert und schließt erstmalig Gleitkommabefehle ein. Daß die Geschwindigkeit gegenüber dem an sich schon sehr schnellen 68000 nochmals gesteigert wurde, liegt auf der Hand. Doch ehe dieser Prozessor auf dem Markt eine große Verbreitung gefunden hat, wird es noch Jahre dauern. Insbesondere der Preis von mindestens 800 DM pro Stück wird die Verbreitung dieses Mikroprozessors nicht gerade fördern. Interessant ist, daß alle Programme für den 68000 auch auf dem 32 Bit Giganten laufen werden. Also ein Grund mehr für die Softwarefirmen, massenweise 68000 Software zu entwickeln, um bei Markteinführung auch schon etwas für den 68020 zu haben!

Außerdem stellt sich die Frage, ob ein 16 Bit Rechner für die meisten Zwecke nicht schon völlig ausreicht. 16 Mbyte Speicher in Verbindung mit hoher Verarbeitungsgeschwindigkeit dürfte selbst den Ansprüchen großer Unternehmen völlig genügen. Wo zum Beispiel durch Multitasking der Datendurchfluß zu klein wird, kann der Arithmetikprozessor 68881 den Kauf eines extrem teueren 32 - Bit - Rechners ersparen.



Als 68020 wird dieser 68000 demnächst einen 32 Bit-Prozessor als großen Bruder bekommen

Fazit: Diese kleine Serie hat wohl sehr deutlich dieVorteile des Motorola 68000 gegenüber seinen Konkurrenten gezeigt. Insbesondere die Forderung nach großem Speicherplatz und hoher Verarbeitungsgeschwindigkeit erfüllt der Motorola 68000 voll und ganz. Alle Anzeichen sprechen daher dafür, daß dieser Typ in den nächsten Jahren auf dem umfangreichen Markt der Mikroprozessoren dominieren wird. ■

Professionelle Software im Preis enthalten

Fortsetzung von Seite 1

Mit Hilfe der gegebenen Mittel ist es kein Problem, ein Kassenbuch oder eine Kostenkalkulation aufzubauen. Eine Liste von 18 Befehlen unterstützt das Arbeiten mit ABACUS. Man kann eine Zelle in einen Bereich von Zellen kopieren, was bei Formeln besonders sinnvoll ist. Der Befehl WINDOW erlaubt es, den Arbeitsbereich in zwei Fenster aufzuspalten, um zwei Bereiche gleichzeitig betrachten zu können. Die Kommandos GRID, DESIGN, JUSTIFY und UNITS verändern das Erscheinungsbild des Arbeitsblattes. Text und Zahlen können rechtsbündig, zentriert oder linksbündig dargestellt werden, das Zahlenformat ist frei wählbar.

Alle Arbeitsblätter können mit SAVE abgespeichert und mit LOAD wieder geladen werden. Wichtiger ist jedoch, daß man mit dem Befehl EXPORT die Daten so abspeichern kann, daß sie von einem der anderen Programme gelesen werden können. So kann eine mit ABACUS erstellte Tabelle mit EASEL graphisch dargestellt werden oder mit QUILL textlich überarbeitet werden.

Abschließend zum ABACUS muß noch folgendes bemerkt werden: In der Grundversion des Sinclair QL stehen dem Programm nur 23 KByte Arbeitsspeicher zur Verfügung. Damit ist es nicht möglich, das gesamte Gitter voll auszunützen, man stößt recht schnell an Speichergrenzen.

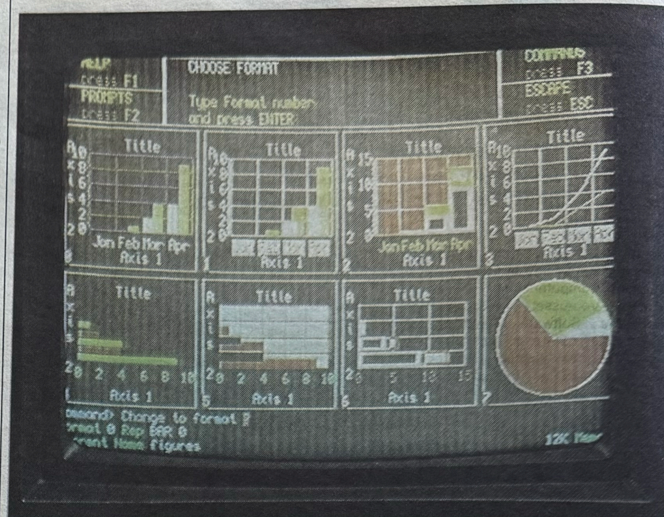
Das Programm EASEL ist ein interaktives Graphikprogramm. Es erlaubt Zahlenketten in Form von Kuchen, Linien oder Balken auf dem Bildschirm darzustellen.

Alle notwendigen Informationen werden vom Programm angefordert und durch das umfangreiche Modul ist die Bedienung von EASEL ein Kinderspiel.

Es gibt mehrere Möglichkeiten, Graphiken einzugeben. Entweder durch direktes Eintippen von Werten, durch Eingabe von Formeln oder durch die Übernahme von Tabellen, die von ABACUS oder ARCHIVE erstellt wurden.

Man kann immer mehrere Graphiken im Speicher halten und diese einzeln oder gleichzeitig in 7 verschiedenen Formaten auf dem Bildschirm darstellen. Zusätzlich sind Hintergrund, Linien und Balken in Form und Farbe aus einer großen vordefinierten Menge frei wählbar oder auch selbst definierbar. In alle Graphiken kann problemlos Text eingestreut werden. Verwendet man zum Erstellen einer Graphik eine Formel, ist es möglich, mathematische Funktionen mit EASEL darzustellen. In den meisten Fällen jedoch wird das Programm wohl dazu benutzt, um Tabellen von ABACUS anschaulich darzustellen. Oft ist es notwendig, eine Hardcopy einer Graphik zu produzieren. Auch das ist möglich, wenn z. B. ein EPSON FX 80 Drucker am QL angeschlossen ist. Andere Drucker können ebenfalls verwendet werden, man muß jedoch vorher den Druckertreiber mit Hilfe einer speziellen Routine an den Drucker anpassen. Eine Alternative zur gedruckten Hardcopy ist die Photographie. Man kann zu diesem Zweck den Kontrollbereich und den Status-Bereich vom Bildschirm löschen, um die Graphik zu vergrößern. Mit einem Farbfilm sind so recht anschauliche Ergebnisse zu erzielen.

Über das Textverarbeitungsprogramm und die Möglichkeiten des Datenbanksystems des QL werden wir in der nächsten Ausgabe von HCR berichten. ■



So sieht das Auswahlmenü von Abakus für die Wahl der darstellenden Grafik aus. Sehr umfangreich und übersichtlich.

MASTERMIND

Spielerklärung: Der Computer stellt dem Spieler eine zu erratende Zahlenkombination (wählbar von 3 bis 6 Zahlen). Die Anzahl der Versuche ist nach Schwierigkeit gestaffelt. Nach jedem Versuch gibt der Computer an, wieviele Zahlen richtig sind und wieviele in der richtigen Position stehen. ■

Von Robert Noppe

Für Schneider CPC 464

```
1 '------- M A S T E R M I N D ------
2 'geschrieben von    Robert Noppe   18.03.1985
3 DIM z(6),x(6)
4 '--------------- Anfangsmaske ----------------
5 CLS:MODE 1:INK 0,2:INK 1,11:BORDER 2
6 PEN 3:PRINT TAB(10) "M A S T E R M I N D"
7 PLOT 48,320:DRAW 608,320:DRAW 608,160:DRAW 48,160:DRAW 48,320
8 PLOT 32,336:DRAW 624,336:DRAW 624,144:DRAW 32,144:DRAW 32,336
9 LOCATE 9,8:PRINT"do you want to guess :"
10 FOR i=3 TO 6
11 LOCATE 6,8+i:PRINT i;"numbers  then  ENTER --->";i
12 NEXT
13 e$=INKEY$:IF e$="" THEN 13
14 IF e$<CHR$(51) OR E$>CHR$(54) THEN 13
15 e=VAL(e$)
16 q=INT(e*2.71)
17 anz=e
18 '--------------- Spielmaske ----------------
19 CLS:MODE 2:INK 0,0:INK 1,25-q:BORDER 0
20 LOCATE 35,1:PRINT"MASTERMIND":LOCATE 35,2:PRINT "=========="
21 LOCATE 3,5:PRINT"your guesses"TAB(30)"Correct numbers"TAB(50)
   "Correct positions"
22 FOR i=1 TO anz
23 s=s+2
24 LOCATE 33+s,3:PRINT CHR$(143)
25 NEXT
26 FOR i=1 TO q
27 t=0
28 FOR x=1 TO anz
29 t=t+3
30 LOCATE t,7+i:PRINT CHR$(233)
31 NEXT:NEXT
32 '--------------- Berechnung ---------------
33 p=0
34 p=p+1
35 v(p)=INT(RND(1)*9)
36 z(p)=INT(RND(v(p))*9)
37 c=0
38 FOR i=1 TO anz
39 IF z(p)<>z(i) THEN 42
40 c=c+1:IF c>1 THEN 41 ELSE 42
41 p=p-1:GOTO 34
42 NEXT
43 IF p<anz THEN 34
44 '--------------- Eingabe ------------------
45 FOR o=1 TO q
46 t=0
47 FOR i=1 TO anz
48 x$(i)=INKEY$:IF x$(i)="" THEN 48
49 IF x$(i)<CHR$(48) OR x$(i)>CHR$(57) THEN 48
50 t=t+3
51 x(i)=VAL (x$(i))
52 PRINT CHR$(&7)
53 LOCATE t-1,7+o:PRINT x(i)
54 NEXT
55 '------------- Correct numbers -------------
56 rt=0
57 r=0
58 r=r+1
59 k=0
60 FOR i=1 TO anz
61 IF z(r)<>x(i) THEN 64
62 k=k+1
63 IF k<2 THEN rt=rt+1
64 NEXT
65 IF r<anz THEN 58
66 LOCATE 35,7+o:PRINT rt
67 '------------ Correct positions ------------
68 b=0
69 FOR i=1 TO anz
70 IF z(i)<>x(i) THEN 72
71 b=b+1
72 NEXT
73 LOCATE 55,7+o:PRINT b
74 IF b=anz THEN 76
75 NEXT
76 '-------------- Spielende -----------------
77 s=0
78 FOR i=1 TO anz
79 s=s+2
80 LOCATE 33+s,3:PRINT z(i)
81 NEXT
82 IF b=anz THEN LOCATE 50,23:PRINT"you win this game" ELSE GOTO 84
83 GOTO 85
84 LOCATE 50,23:PRINT"you lost this game"
85 LOCATE 40,25:PRINT"do you want to play again (y/n)"
86 i$=INKEY$:IF i$="" THEN 86
87 IF i$="n" OR i$="N" THEN END
88 IF i$="y" OR i$="Y" THEN RUN 3:GOTO 86
```

— TEXTVERARBEITUNG —

für Schneider im Vergleich

Von F. Lorenz

Der Schneider CPC 464, Renner auf dem Home - Computer - Sektor, könnte so manchem Personal Computer Konkurrenz machen - Voraussetzung die angebotene Software stimmt. Denn die Qualität von Tastatur und Bildschirmdarstellung lassen professionelles Arbeiten zu.

Die Stunde der Wahrheit kommt meist mit dem Textverarbeitungsprogramm. HCR hat einige kassettenorientierte Programme getestet. Das Ergebnis vorweg: Man nehme von jedem Programm etwas und lasse sich ein neues einfallen.

Topword / Easi Amsword

Da ist zunächst mal das Schneider - Programm ,,Topword / Easi Amsword''. Problemlos zu bedienen, ist es für Viel- und Schnellschreiber recht geeignet. Es wird ganz einfach im 40- oder 80 - Zeichen - Modus drauflosgeschrieben. Einfügungen sind jederzeit und überall im Text möglich. In der letzten Bildschirmzeile wird man stets über die aktuelle Zeilen- und Anschlagszahl sowie über den Schreibmodus (Einfügen/Überschreiben) und den File - Namen unterrichtet.

Die Menüsteuerung sieht auch eine Farbwahl vor, die vier festgelegte Kombinationen zuläßt und ausreichend ist. Gespeichert werden kann wahlweise mit 1000 oder 2000 Baud, wobei die Bildschirmanweisung etwas unklar ist und anfänglich zu Bedienungsfehlern führen kann. Wer allerdings ein paar mal REC und PLAY am Recorder zu spät gedrückt hat, wird es bald nicht mehr tun.

Von Vorteil kann sein, daß unter dem Menüpunkt ,,Bearbeiten'' der Name des gerade am Schirm stehenden Textes geändert werden kann. Braucht man alles, was auf einer Kassette gespeichert wurde, mal auf Papier, so ist man keineswegs gezwungen, File für File zu laden und übers Druckermenü auszugeben. Die Funktion ,,Cassette drucken'' spuckt den gesamten Kassetteninhalt ohne weiteres Zutun auf den Drucker.

Keine Textparameter

Ausgesprochen schlecht ist hingegen, daß keinerlei Textparameter eingegeben werden können, also formatiertes Schreiben von beispielsweise Geschäftsbriefen, am Bildschirm Illusion bleibt. Über die Notwendigkeit von Blockoperationen kann man streiten, hier fehlen sie jedenfalls. Auch fehlt dem Programm der deutsche Zeichensatz, der allerdings wenigstens gesondert angeboten wird (DEUTAST). Ein weiteres Manko ist die fehlende Silbentrennung. Dadurch verliert die Möglichkeit, im Blocksatz auszudrucken, völlig ihren Reiz. Die dabei entstehenden Löcher im Schriftbild sind mitunter doch allzu groß.

Lob ist dem reichhaltigen Druckermenü zu spenden. So kann mit bis zu zehn ASCII - Steuerzeichen der Drucker programmiert werden. Seitenlänge und -breite ist einstellbar, wobei praxisgerecht auch der linke Rand programmiert werden kann. Auf Wunsch kann fortlaufend die Seitenzahl gedruckt werden. Auch der Seitenvorschub ist je nach angeschlossenem Drucker einstellbar. Dazu kommt noch ein- oder zweizeiliges Drucken und die Angabe, wieviele Kopien von dem Text gedruckt werden sollen.

Alles in allem: Schade um die fehlende Möglichkeit des formatierten Schreibens.

CPC - Text / Adreß

Wesentlich besser bedient wird man in diesem Punkt mit dem CPC - Text/Adreß aus dem Markt + Technik Verlag, Haar b. München. Der integrierte deutsche Zeichensatz macht das CPC - Keyboard zu einer echten Scheibmaschinentastatur. Alle deutschen Sonderzeichen finden sich dort, wo sie auch hingehören; kein fingerbrechender Umweg über die eckigen Klammern. Wird der Epson - kompatible Zeichensatz vom Drucker nicht geschluckt, kann er umdefiniert werden; die Bedienungsanleitung gibt darüber Auskunft.

Die Textparameter können innerhalb eines Textes jederzeit geändert werden. Auch umformatieren auf eine andere Zeilenbreite und Seitenlänge ist stets möglich. Die Trennfunktion, die zwar nur halbautomatisch und absatzweise abläuft, macht den mit automatischem Word - Wrapping geschriebenen Text zu einem lesbaren Blocksatz. Bei langen Texten fällt allerdings die Langsamkeit der Silbentrennung stark auf. Man kann dabei mühelos den Text nochmal durchlesen und gegebenenfalls ändern.

Die Cursorsteuerung erlaubt ebenfalls komfortables Arbeiten. Wichtig für eine gute Textverarbeitung ist auch die Möglichkeit des wortweisen Löschens. In einer verbesserten Version sind auch alle Blockoperationen möglich.

Zeitraubendes Ladeverfahren

Etwas umständlich ist die Datacorder - Steuerung. Beim ,,load'' - Vorgang fordert das Programm keinen File - Namen an, sondern arbeitet das Band einfach ab, indem es jedesmal fragt, ob der gefundene Text geladen werden soll. Also ein reichlich zeitraubendes Verfahren, wenn man vergessen hat, Textname und vor allem Zählwerkstand zu notieren.

Die erste Version von ,,CPC - Text/Adreß'' weist ein recht dürftiges Druckermenü auf. So kann beispielsweise nur die Seitenlänge eingegeben oder über den Programmpunkt ,,Massenbrief'' auf den Adreßteil zugegriffen werden. Auch ist in der Ur - Form kein linker Randsteller vorgesehen, so daß auf softwaregesteuerten Druckern praktisch nichts geschrieben werden kann, was später abgeheftet werden muß.

In der verbesserten Form läßt sich der linke Rand dann ebenso beliebig einstellen wie die Zeilenbreite schon bisher. Auch verschiedene Druckersonderfunktionen, wie beispielsweise Fettsatz, Breitschrift, Schmalschrift, drei verschiedene Zeilenabstände können beliebig direkt während des Schreibens angewählt und auch mitten in der Zeile geändert werden. Außerdem lassen sich die Steuercodes für den Drucker auch ändern. Insgesamt sind elf verschiedene Druckerbefehle möglich, wobei allerdings vier für Ausschaltkommandos gebraucht werden. Aufgerufen werden die Druckerbefehle mit CTRL/Zifferntasten.

Auch der Umstand, daß ein vorhandener Text nicht mit einem Tastendruck gelöscht werden kann, gehört in der überholten Version der Vergangenheit an. Hinzugekommen ist auch, daß das Programm die Diskettenstation unterstützt, man also wahlweise per Menü beispielsweise von Diskette oder Kassette laden oder eine Disc - Kopie ziehen kann. Im übrigen gibt es das CPC - Text / Adress auch auf Diskette.

Alle Farben vorhanden

Anzumerken wäre noch, daß bei der Farbwahl aus der gesamten Palette des CPC geschöpft werden kann. Was den Adreßteil der Kassette betrifft, so gefallen vor allem die Selektionskriterien, die sich auch untereinander kombinieren lassen und so den Druck gezielter Massenbriefe mit automatischem Einlesen der Adressen erlauben.

Einlesen muß man sich im übrigen erst mal in die 16seitige Bedienungsanleitung (1. Version), die sprachlich und graphisch alle Funktionen erklärt. Die Textgestaltung aber erschwert das Finden einzelner Funktionsbeschreibungen.

Tasword 464

Mit einem 45 Seiten starken Handbuch wartet ,,Tasword 464'' auf. Einige Englischkenntnisse sind erforderlich. Um dem Tasword - User ständiges Blättern zu ersparen, ist das komplette Programm-Menü jederzeit auf dem Bildschirm abrufbar. Ein zuladbarer Tutor präsentiert sogar das Handbuch.

Alle Möglichkeiten dieses englischen Programms zu beschreiben, würde den Rahmen sprengen. Variable Textformatierung und wahlweiser automatischer Zeilenumbruch ist selbstverständlich. Etwas langsam läuft die Such-/Ersetzfunktion ab. Mehrere Texte lassen sich zusammenhängen. Positiv auch: Tasword verstand auch

Kompatibilität zu Textdateien anderer Programme ist gewährleistet

mit anderen Programmen erstellte Texte und verarbeitete sie wie eigene.

Um nicht immer wieder von neuem alle gewünschten Text- und Druckparameter eingeben zu müssen, kann die einmal erstellte Fassung abgespeichert werden. Noch ein Wort zur Speicherfunktion: Per Zugriff zum Basic - Modus können Speicher - Fehler erkannt werden. Wird eine ,,Error'' - Meldung ausgeworfen, kann ,,save'' wiederholt werden.

Auch Tasword hat Nachteile

Trotz aller Vorzüge, hat Tasword 464 einige Nachteile. So fehlt der deutsche Zeichensatz, der auch nicht dazugeladen werden kann. Dagegen kann man die fehlende Silbentrennung fast verschmerzen. Bei ZS - Soft in Berchtesgaden ist jedoch eine deutsche Fassung in Arbeit.

Hinderlich bei allen Programmen ist eine Eigentümlichkeit des CPC, die garbage collection. Das Basic - System ordnet damit den Speicher. Für etliche Sekunden wird dann der Bildschirm ,,eingefroren''. Je länger ein Text wird, umso häufiger tritt dieser Zustand auf und umso länger dauert er. Zwar kann man sozusagen blind weiterschreiben, doch Schnellschreiber werden immer wieder feststellen, daß der CPC manche Buchstaben verschluckt.



Das Bild zeigt den Startscreen von unserem Textprogramm Microtex aus Ausgabe 7/85

Das könnte besser sein
Diesmal Schneider CPC

Textverarbeitung zählt sicher zu den Problemgebieten der Computerei. Das Ei des Kolumbus hat sicher noch niemand gefunden.

Die Ansprüche der Anwender und die Vorstellungen der Entwickler sind auf beiden Seiten viel zu differenziert, als daß sie allgemeingültig unter einen Hut gebracht werden könnten. Das gilt nicht nur für die Einsteiger - Klasse, sondern auch für 30 000 Mark - und mehr - Anlagen.

Bei dem heutigen Standard ist allerdings nicht einzusehen, warum die beste Software nichts mehr nützt, wenn der Strom ausfällt. Jeder kleine Hand - held - Computer behält seinen Text, wenn man ihn abschaltet. Die Home - Computer - Klasse mit PC - Ambitionen versagt dagegen kläglich.

Dies sei vor allem dem ob seiner Qualitäten gerühmten CPC 464 angekreidet. Gar nicht PC - like ist seine 5 - Volt - Stromversorgung weder durch eine Batterie noch durch einen Akku gepuffert. Dabei ist selbst für einen Bastler diese Problemlösung eine der leichtesten.

Auch könnte Schneider seinem Amstrad längere Nabelschnüre zwischen Monitor und Tastatur spendieren.

Wer viel Textverarbeitung betreibt, mag den Schirm nicht direkt vor der Nase. Aber auch hier bleibt nichts anderes, als der Griff zum Lötkolben. Probleme mit Spannungsabfall oder Störeinstrahlung entstehen dabei selbst bei Zwei - Meter - Verlängerungen nicht.

Auch das serienmäßige Druckerkabel ist letztlich zu kurz geraten - zwei Meter wären auch hier angebracht und datentechnisch unproblematisch. Abhilfe: Do - it - yourself.

Auf den Nenner gebracht: Ein Home - Computer, der in seinem Namen nicht ganz zu Unrecht den PC herauskehrt, sollte Handgeschneidertes nicht nötig haben und den Aufbau eines ergonomischen Arbeitsplatzes erlauben.

fjl

Directory drucken

Von Peter Eckhoff

Jeder kennt den Ärger, wenn man seine Disketten auf Vordermann, bzw. seine Programme sortieren will. Directory laden / OPEN 1,4:CMD1:LIST/ nach dem Ausdruck mit CLOSE 1 oder RUN-STOP/RESTORE den Drucker wieder abschalten und die nächste Directory laden. Vergißt man etwas, bekommt man entweder recht merkwürdige Zeichen ausgedruckt oder eine Fehlermeldung z. B.: FILE OPEN serviert. Das vorliegende Programm liest das Inhaltsverzeichnis einer Diskette ein und gibt es wahlweise auf einen Drucker oder den Bildschirm aus. Dabei läßt sich der Drucker beliebig vom Programm aus einstellen. Der Teil des Programms „DRUCKERMARMETER EINST. muß natürlich von jedem Anwender auf seinen eigenen Drucker, bzw. die Möglichkeiten seines Druckers abgeändert werden. Ein Compilieren des Programms z. B. mit Petspeed bringt erhebliche Zeitersparnis beim Lesen der Directory.

In Zeile 70 werden die Variablen B1$/C1$/D1$ mit 144 dimensioniert, da mehr Einträge in die Directory nicht möglich sind. Springt man vom Menue (Zeile 100-290) zum Punkt LESEN, dann wird zuerst eine eventuell noch vorhandene Directory gelöscht (Zeile 320-380). Dann liest das Programm die neue Directory ein (Zeile 460-640). Wem diese Leseroutine bekannt vorkommt, dem sei gesagt, daß sie dem Programm 'DIR' (Listing im Floppyhandbuch) entnommen ist. Nur werden die gelesenen Daten nicht auf den Bildschirm ausgegeben, sondern in den in Zeile 70 dimensionierten Variablen gespeichert. Diese Variablen werden dann später bei der Bildschirm- (Zeile 640-830) oder Druckerausgabe (Zeile 840-1000) ausgegeben. Bei der Druckerausgabe muß jedoch die entsprechende Druckzeile formatiert, bzw. alle Zeilen gleich lang gemacht, werden. (Dies geschieht in den Zeilen 970 - 1000).

Die ab Zeile 1010 einstellbaren Druckerparameter erlauben ein Ausdrucken in verschiedenen Schriftarten. Druckt man z.B. komprimiert (17 cpi) dann passen auch einmal zwei Ausdrucke nebeneinander auf die Diskettenhülle. Dies ist bei längeren Directory's recht nützlich.

— Variablentabelle —

A$,B$,C$,C - dienen als Zwischenspeicher beim Lesen von Daten
B1$(,C1$(,D1$(- speichern die Directoryeinträge
Q - Anzahl der Einträge
F$ - füllt alle Einträge bis auf die gleiche Länge auf
P,P$ - dienen zum Einstellen von Druckerparametern
ST - Statusvariable zum Erkennen des Dateiendes
T - Schleifenvariable für die Ausgabe der Daten.

```
10 REM ********************
20 REM * DIRECTORYDRUCKER *
30 REM *                  *
40 REM *     (C) BY:      *
50 REM *  PETER ECKHOFF   *
60 REM ********************
70 DIM B1$(144),C1$(144),D1$(144)
80 POKE53280,1:POKE53281,1:PRINT"█";
90 F$="                    "
100 REM *********
110 REM * MENUE *
120 REM *********
130 PRINT"███                                          ██";
140 PRINT"█                                            █";
150 PRINT"█                                            █"
160 PRINTTAB(5);"█████┌──────────────────┐"
170 PRINTTAB(5);"██│DIRECTORYDRUCKER│"
180 PRINTTAB(5);"█└──────────────────┘█ (C)BY:P.ECKHOFF██"
190 PRINT"        ██┌──────────────┐█"
200 PRINT"        █│BITTE WAEHLEN:│█"
210 PRINT"        █└──────────────┘████"
220 PRINTTAB(5);"█1█ = DIRECTORY LESEN██"
230 PRINTTAB(5);"█2█ = DIRECTORY ZEIGEN██"
240 PRINTTAB(5);"█3█ = DIRECTORY DRUCKEN██"
250 PRINTTAB(5);"█4█ = DRUCKER EINSTELLEN██"
260 PRINTTAB(5);"█5█ = PROGRAMMENDE"
270 GETA$:IFA$=""THEN270
280 IFASC(A$)<49ORASC(A$)>53THEN270
290 ON VAL(A$) GOTO 330,670,870,1040,1250
300 REM **********************
310 REM * ALTE DIR. LOESCHEN *
320 REM **********************
330 FORT=1TOQ
340 C1$(T)=""
350 B1$(T)=""
360 D1$(T)=""
370 NEXTT
380 Q=1
390 REM ****************
400 REM * LESEROUTINE *
410 REM ****************
420 POKE53280,0:POKE53281,0:PRINT"███████████"
430 PRINTTAB(14);"█***************█"
440 PRINTTAB(14);"█*BITTE WARTEN*█"
450 PRINTTAB(14);"█***************█"
460 OPEN1,8,0,"$0"
470 GET#1,A$,B$
480 GET#1,A$,B$
490 GET#1,A$,B$
500 C=0
510 IF A$<>"" THEN C=ASC(A$)
520 IF B$<>"" THEN C=C+ASC(B$)*256
530 D1$(Q)=MID$(STR$(C),2)
540 GET#1,B$:IF ST<>0 THEN 620
550 IF B$<>CHR$(34) THEN 540
560 GET#1,B$:IF B$<>CHR$(34)THEN B1$(Q)=B1$(Q)+B$:GOTO560
570 GET#1,B$:IF B$=CHR$(32) THEN 570
580 C$=""
590 C$=C$+B$:GET#1,B$:IF B$<>""  THEN 590
600 C1$(Q)=LEFT$(C$,3)
610 IF ST=0 THEN Q=Q+1:GOTO480
620 B1$(Q)="BLOCKS FREE"
630 CLOSE1:PRINT"█":GOTO80
640 REM ********************
650 REM * BILDSCHIRMAUSGABE *
660 REM ********************
670 PRINT"██"
680 FORT=1TOQ
690 PRINTD1$(T)
700 IFT=1THENPRINT"█";
710 PRINT"█";TAB(4);CHR$(34)
720 IFT=1THENPRINT"█";
730 PRINT"█";TAB(5);B1$(T);CHR$(34)
740 IFT=1THENPRINT"█";
750 PRINT"█";TAB(23);C1$(T)
760 IFT=1THENPRINT
770 NEXTT
780 REM
790 PRINT"█ ███IEDERHOLEN   ███RUCKEN   ███ENUE"
800 GETA$:IFA$="W"THEN670
810 IFA$="D"THEN870
820 IFA$="M"THEN80
830 GOTO800
840 REM ******************
850 REM * DRUCKERAUSGABE *
860 REM ******************
870 OPEN4,4,0
880 POKE53280,5
890 GOSUB970
900 FORT=1TOQ
910 IFT=1THENPRINT#4,CHR$(18);
920 PRINT#4,LEFT$(D1$(T),4);CHR$(34);LEFT$(B1$(T),16)
    ;CHR$(34);LEFT$(C1$(T),3);
930 IFT=1THENPRINT#4,CHR$(146);
940 PRINT#4
950 NEXTT
960 CLOSE4:GOTO80
970 FORT=1TOQ
980 C1$(T)=C1$(T)+F$:D1$(T)=D1$(T)+F$:B1$(T)=B1$(T)+F$
990 NEXTT
1000 RETURN
1010 REM ***************************
1020 REM * DRUCKERPARAMETER EINST. *
1030 REM ***************************
1040 PRINT"██      DRUCKER PARAMETER EINSTELLEN"
1050 PRINT"█       ──────────────────────────██"
1060 OPEN1,4,1
1070 PRINT"█  1.SCHRIFTART"
1080 PRINT"█  ────────────█"
1090 PRINT"█  1=PICA    2=ELITE    3=KOMPRIMIERT██"
1100 GETP$:IFP$=""THEN1100
1110 IFASC(P$)<49ORASC(P$)>51THEN1100
1120 IF VAL(P$) = 1 THEN P=78
1130 IF VAL(P$) = 2 THEN P=69
1140 IF VAL(P$) = 3 THEN P=81
1150 PRINT#1,CHR$(27)CHR$(P)
1160 PRINT"█  2.ZEILENABSTAND"
1170 PRINT"█  ───────────────█"
1180 PRINT"█  1 = 1/6 ZOLL        2 = 1/8██"
1190 GETP$:IFP$=""THEN1190
1200 IFASC(P$)<49ORASC(P$)>50THEN1190
1210 IF VAL(P$) = 1 THEN P=65
1220 IF VAL(P$) = 2 THEN P=66
1230 PRINT#1,CHR$(27)CHR$(P)
1240 CLOSE1:GOTO80
1250 POKE53280,254:POKE53281,246:PRINT"███
     (C) BY: P.ECKHOFF██":END
```

Telebuilding-Set: 2 D im BASIC

Telebuilding ist eine Lernmethode zum Erstellen von Telespielprogrammen für den VC 20 in BASIC. Durch verschiedene Kombinationen der einzelnen „Generatoren untereinander" und einer regen Phantasie des jeweiligen Programmierers entwickeln Sie ganz leicht ein eigenes Telespiel.

Von Andreas Lendlein

Warum gibt es Telebuilding-Set 2D?

Hinter TBS steckt die Idee, dem BASIC-lernenden relativ schnell die Möglichkeit zu geben, durch eigene Kreativität Programme, speziell Telespiele, zu gestalten. Konnte man BASIC bisher noch nicht korrekt, tippte man einfach stumpfsinnig Listings aus Büchern oder Zeitschriften ab. Man verstand zwar nicht, wie es funktionierte, aber Hauptsache, man hatte ein „eigenes" Telespiel. Bei TBS ist dies anders. Sie sollen verstehen, was Sie da abtippen. Verschiedene Generatoren (Unterprogramme) z. B. zur Bewegung eines vom Computer gesteuerten Raumschiffes werden durch die Steuerung (Hauptprogramm) z.B. zur Bewegung des vom Spieler gesteuerten Raumschiffes, verbunden. Mit ein bißchen Grafik (beispielsweise einem neuen Zeichensatz) und einem guten Sound läßt sich auf diese Weise dann ein ganz ordentliches Programm zusammenstellen.

Was benötigt man zu TBS?

Sie benötigen Grundkenntnisse in BASIC, einen BASIC-kompatiblen Microcomputer (in unseren Beispielprogrammen wurde ein VC 20 mit 32 kBytes-Erweiterung verwendet) und ein wenig Geduld. Um die Arbeitszeit etwas zu verkürzen, gibt es Tools. Tools (deutsch: Werkzeug) wären z. B. eine Renumber- oder Autofunktion, doch diese will ich hier nicht behandeln, da es schon eine relativ große Anzahl von Lösungen dieser Probleme gibt. Ich suchte einen Ersatz für die Key-Funktion und zwar möglichst in BASIC. Wer hat sich nicht schon darüber geärgert, daß sein VC 20 zwar 8 frei belegbare Funktionstasten besitzt, diese aber nicht richtig ausnutzen kann. Mein Lösungsvorschlag gibt die Möglichkeit, während des Programmierens die Tasten F 1 - F 8 mit längeren Wörtern oder Teilprogrammzeilen, die öfters benutzt werden, zu belegen. Leider können dazu keine Variablen, die zuvor durch eine Eingaberoutine komfortabel jeweils belegt würden, benutzt werden, da das Programm bei Bestätigung der RETURN-Taste immer wieder unterbrochen und dann mit RUN gestartet wird. Mit RUN würden dann auch gleichzeitig alle Variablen gelöscht. Um das Programm automatisch wieder zu starten, wird der Tastaturpuffer verwendet. Die KEY-

```
1 REM"███████*************************
2 REM"███████*                       *
3 REM"███████*   GALACTIC -ELSTER    *
4 REM"███████*                       *
5 REM"███████* VON   A.LENDLEIN      *
6 REM"███████*                       *
7 REM"███████*   JANUAR    1985      *
8 REM"███████*                       *
9 GOTO300:REM"████████████████*************************
10 REM VORPRG.
20 FORI=0TO2047:POKE5120+I,PEEK(32768+I):NEXT
30 FORT=0TO7:POKE6400+T,255:NEXT
40 POKE4096,66
50 H=7392-2048
60 FORT=1TO10
70 FORP=0TO7:READY:POKEH+P,Y:NEXTP
80 H=H+8:NEXTT
90 DATA0,24,60,126,126,255,255,66
100 DATA24,60,195,66,60,90,126,255
110 DATA0,0,255,255,255,68,68,238
120 DATA255,255,255,255,255,255,255,255
130 DATA0,0,0,0,0,0,0,0
140 DATA192,248,240,252,128,192,0,224
150 DATA0,0,96,60,60,96,0,0
160 DATA24,60,126,0,90,90,60,60
170 DATA3,6,15,94,254,15,6,3
180 DATA127,31,7,3,3,1,1,0
190 H=7640-2048
200 FORT=1TO5
210 FORP=0TO7:READY:POKEH+P,Y:NEXTP
220 H=H+8:NEXTT
223 PRINT"██";
225 LOAD"ANL*",8
230 DATA56,124,84,108,56,68,56,0
240 DATA3,6,15,94,254,15,6,3
250 DATA31,7,15,0,3,63,15,3
260 DATA192,96,240,122,127,240,96,192
270 DATA254,252,248,248,240,192,128,0
300 REM ANFANGSGRAPHIK
305 PRINT"█";
310 POKE36879,8
320 A$="███                    ██"
330 B$="██ ██                 ███"
340 C$="██  ██               ████"
350 D$="██   █              ██ ██"
400 PRINTA$;B$;C$;
410 FORT=1TO17
420 PRINTD$;
425 NEXT
430 A=4096+22*20
440 FORT=0TO1
450 POKEA+T,160:POKEA+T+33792,6
460 NEXT
470 IFB=0THENB=1:A=A+20:GOTO440
480 A=A+2
490 POKEA,160:POKEA+33792,6
500 A=A+21:POKEA,160:POKEA+33792,6
510 A=A-1:POKEA,95:POKEA+33792,6
520 A=A-23:POKEA,95:POKEA+33792,6
530 A=A+46:POKEA,95:POKEA+33792,6
540 A=A-21:POKEA,105:POKEA+33792,6
550 A=A-21:POKEA,105:POKEA+33792,6:B=B+1
560 IFB=2GOTO550
570 PRINT"█████";
580 PRINT"█████ ████████████ ████";
590 PRINT"█████████████████████████";
600 PRINT"█████████████████████████";
610 PRINT"███";
620 PRINT"█████ ██████████ ████";
630 PRINT"██████ ██████████ ████";
640 PRINT"█████████████████████████";
650 PRINT"██████* PRAESENTIERT *████";
660 GOSUB1000
670 PRINT"██████ GALACTIC -█████████";
680 PRINT"██████          ELSTER"
685 GOSUB2000
690 GOTO10
1000 REM UNTERPRG.
1005 FORU=1TO5
1010 FORT=0TO15
1020 C=16*T+8
1025 POKE36879,C
1026 FORI=1TO150:NEXT
1030 NEXT
1034 NEXTU
1035 POKE36879,8
1040 RETURN
2000 REM UNTERPRG.
2010 FORT=1TO100
2020 FORI=0TO7
2030 POKE36879,I OR 10
2035 REM VERZOEGERUNG DURCH REMARK-ZEILE
2036 REM ZWEITE VERZOEGERUNG
2040 NEXTI
2050 NEXTT
2060 RETURN
```

Fortsetzung im nächsten Heft

```
10 REM ANLEITTUNG
20 FORT=1TO25:PRINT"■";:FORI=1TO180:NEXT:NEXT
30 PRINT"■";
40 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■";
45 FORI=0TO350:READA$:PRINTA$;:FORT=1TO30:NEXT:NEXT
46 GOSUB1000:POKE36879,8
47 LOAD"HAU*",8
50 DATA"          A",N,L,E,I,T,U,N,"G      "
60 DATA"          -","-","-","-","-","-","-","-         "
70 DATA"■SIE■"," ",S,T,E,U,E,R,N," ",E,I,N,E," "
80 DATAG,A,-,"  ",L,A,C,T,I,C,-,E,L,S,T,E,R,.,"      "
90 DATAB,E,S,C,H,U,E,T,Z,E,N," ",S,I,E," ",I,H,R,"   ",N,E,S,T,.
100 DATA"                    ",A,B,E,R," ",V,O,R,S,I,C,H,T," ",V,O,R," ",D,E,N,"
110 DATAH,U,M,A,N,O,I,D,E,N
120 DATA.,"          "
130 DATAS,O,L,L,T,E," ",E,S," ",D,I,E,S,E,N," ",G,E,-,"  "
140 DATAL,I,N,G,E,N," ",A,L,L,E," ",","," ",I,H,R,E,"  "
150 DATAS,C,H,A,E,T,Z,E," ",Z,U," ",R,A,U,B,E,N,"   ",O,D,E,R
160 DATA" ",I,H,R,E," ",3," ",R,A,U,M,S,C,H,I,F,-,F,E
170 DATA" ",Z,U," ",V,E,R,N,I,C,H,T,E,N," ,",D,A,N,N
180 DATAI,S,T," ",E,S," ",A,U,S," ",M,I,T," ",I,H,N,E,N," ."
190 DATAF,U,E,R," ",J,E,D,E,N," ",V,E,R,N,I,C,H,T,E,"- T",E,N
200 DATA" ",A,N,G,R,E,I,F,E,R," ",E,R,H,A,L,T,E,N
210 DATA" ",S,I,E," ",P,U,N,K,T,E," .          "
220 DATAH,A,B,E,N," ",S,I,E," ",G,E,N,U,G," ",P,U,N,K,T,E
230 DATAE,R,R,E,I,C,H,T," ",W,I,R,D," ",D,E,R," ",L,E,"- "
240 DATAV,E,L," ",U,M," 1 ",E,R,H,O,E,H,T," ",U,N,D,"■"
1000 PRINT"■■■";:PRINT"■■■BITTE EINE TASTE"
1010 POKE36879,9:::::::::::::GETA$:POKE36879,8:::IFA$<>""THENRETURN
1020 GOTO1010
```

Anleitung

```
10 PRINT"■■■"
20 POKE44,28:POKE7168,0
25 LOAD"XXX*",8
```

Loader

Funktion belegt die ersten 15 Programmzeilen, die Unterprogramme liegen über der Zeilen-Nr. 63000. Es ist daher nötig, darauf zu achten, beim Programmieren frühestens mit der Zeile 16 zu beginnen oder spätestens mit 62999 aufzuhören. Für den Normalfall dürften ja rund 6200 Programmzeilen genügen. Leider muß bei Benutzung des Hilfsprogramms auf den Cursor verzichtet werden, da sonst wiederum eine Maschinenspracheroutine die Rettung wäre.

START
↓
(Titelbild) — 1. Teilprogramm
↓
(neuer Zeichensatz)
↓
Programm
(Spielanleitung)
(Zeichenerklärung) — 2. Teilprogramm
↓
Spielfeldaufbau — 3. Teilprogramm
↓
Hauptprogramm

— Die Struktur von TBS —

```
0 X=1024:A=0:C=0:REM VARIABLEN
1 K=0
5 GOTO200
10 REM STEUER
30 EX=X:A=PEEK(197):REM TASTATURABFRAGE
40 IFA=10THENX=X-1:REM LINKS
50 IFA=18THENX=X+1:REM RECHTS
60 IFA= 9THENX=X-40:REM HOCH
70 IFA=23THENX=X+40:REM RUNTER
80 IFX<1024THENX=1024:REM ZU HOCH
90 IFX>1024+25*40THENX=1024+25*40:REM"■■■■■ZU TIEF
100 IFEX<>XTHENPOKEEX,32:REMALTE POSITION LOESCHEN
110 POKEX,50:REM NEUE POSITION ZEICHNEN
120 RETURN:REM ZURUECK ZUM TAKTER
200 REM TAKTGEBER
210 IFC=100THENC=0
220 C=C+1
230 IFINT(C/10)=C/10THENGOSUB10:REM SPRINGE ZUR STEUERUNG
240 IFC=20 ORC=[illegible]0THENGOSUB500:REM VERZWEIGUNG ZUR COMPUTERSTEUERUNG
300 GOTO200
500 REM VOM COMPUTER GESTEUERT
510 IFK=1THEN530:REM IST SCHON EIN RAMSCHIFF IN BETRIEB ?
520 G=INT(RND(1)*39)+1024:K=1:REM NEUE STARTPOSITION
530 EG=G
540 G=G+40:REM EINS WEITER NACH UNTEN
550 IFG[illegible]023THENK=0:POKEEG,32:RETURN:REMWENN ZU WEIT UNTEN ,DANN NEUES RAUMSCH.
560 POKEEG,32:POKEG,40:REM ALTE POSITION LOESCHEN,NEUE ZEICHNEN
570 RETURN:REM ZURUECK ZUM TAKTER
```

Beispiel-Programm

```
0 REM FUNKTIONSTASTEN
1 POKE53280,0:POKE53281,0:PRINT"■":GOTO3:REM VC - 20 = POKE36879,8
2 PRINT"■          ■■■■":PRINT"■■          ■■■■":GOTO1
3 WAIT198,1:GETB$:PRINTB$;
4 IFB$=CHR$(13)THENGOSUB63000
5 IFB$="■"THENGOSUB63050
6 REM F2....USW.
10 GOTO3
63000 PRINT"■RUN2":FORI=631TO634:POKEI,145:NEXT:POKE635,13
63001 POKE636,13:POKE198,6:END
63050 PRINT"PRINT ":RETURN
```

Funktionstasten

Aber das Programm sollte ja sowieso in BASIC verfaßt sein.

TBS Das Prinzip

Die wichtigste Sache, bevor man überhaupt beginnt, ein Telespiel zu programmieren, ist selbstverständlich die Spielidee. Diese möchte ich aber erst später abhandeln, um zunächst das Prinzip von TBS zu erklären.

Ein Telespiel in TBS hat eine bestimmte Struktur: (fakultative Teile sind eingeklammert).

Erläuterung:

Das Thema stellt den zeitlichen Ablauf eines Telespiels dar. Start: hier wird z. B. der BASIC-Zeiger verschoben, wenn ein VC 20 mit 32 k-Erweiterung benutzt wird, um den neuen Zeichensatz zu schützen. Titelbild: Eine Bildgrafik (für die normalerweise die Grafiksymbole des Computers ausreichen) mit Spieltitel und Verfassernamen. Hier ist der eigenen Kreativität keine Grenze gesetzt. Neuer Zeichensatz: Um beispielsweise ein Auto über den Bildschirm fahren zu lassen, besitzt der VC 20 kein Zeichen. Also wird ein neues definiert. Spielanleitung (instructions): Auch aus einer Spielanleitung kann man optisch einiges machen. Eine Möglichkeit wäre der Schreibmaschineneffekt.

Spielfeldaufbau: Hier wird die Hintergrundgrafik für das eigentliche Spiel erstellt. Hauptprogramm: Hier erst beginnt das wirkliche Spiel. Doch darüber später mehr. Da die Speicherkapazität besonders bei der Grundversion nur sehr begrenzt ist, teilt man das Spiel möglichst in einzelne Teilprogramme, die nacheinander von der Datasette oder der Diskette geladen werden. Eine zweite Lösung wäre das Weglassen der fakultiven Programmteile unter Beibehaltung des Kernprogrammes.

Das Hauptprogramm

Grundsätzlich kann der Computer auf dem Bildschirm nie zwei Dinge genau gleichzeitig bewegen. Bewegungen, die für das Auge gleichzeitig ablaufen, führt der Computer in Wirklichkeit unmittelbar hintereinander aus. Deshalb ist das Kernstück des Hauptprogrammes ganz einfach ein Zählwerk, das ununterbrochen von z. B. 1 bis 100 zählt. Das sieht dann wie folgt aus:

10 IF A = 100 THENA = 0
20 A = A + 1
30 GOTO 10

Die Zeilen zwischen 20 und 30 dienen dazu, zu den einzelnen Unterprogrammen (Generatoren) zu verzweigen. Das kann natürlich mehr oder weniger oft zum gleichen geschehen:
oft: IF A = 2 x OR A = 40 OR A = 60 THEN
GOSUB ......GENERATOR
nur einmal pro Zähleinheit:
IFA = 5 0 THEN GOSUB ... GENERATOR.

Der Hauptgenerator

Unter dem Hauptgenerator versteht man das Unterprogramm, das dazu dient, den Gegenstand, den später der Spieler steuert, zu bewegen. Dazu wird hier eine Tastaturabfrage benötigt. (Den Joystick möchte ich außer acht lassen) Die Tastaturabfrage besteht entweder aus einem GET-Befehl oder, was besser ist, aus der Abfrage PEEK(197). Die Position auf dem Bildschirm bestimmt der Computer mit Hilfe des Bildschirmspeichers (C 64: ab 1 0 24; VC 20 GV: ab 7680; VC 20 8k: ab 4096). Nennen wir den momentanen Standpunkt des Gegenstandes X. Wollte man diesen um 1 Kästchen nach oben bewegen, bedeutet das für den C 64 X = X - 4 0; für den VC 20 X = X-22 usw.

Die Generatoren

Die übrigen Generatoren bewegen die allein vom Computer träge gesteuerten Gegenstände auf dem Bildschirm. Diese Generatoren sind ähnlich dem Hauptgenerator aufgebaut. Nur entfällt hier die Tastaturabfrage. Dafür wird eine Abfrage eingesetzt, ob bereits ein Gegenstand dieses Generators auf dem Bildschirm ist. Stellen wir uns einen Generator vor, der einen Gegenstand senkrecht von oben nach unten über den Bildschirm fahren läßt. Dabei ist die Stelle auf dem Bildschirm jedesmal verschieden. (C 64: 4 0 Möglichkeit: VC 20 : 22 Möglichkeiten).

Der Gegenstand beginnt sich zu bewegen. R ist die Variable, die angibt, ob gerade ein Gegenstand bewegt wird. Ist K = 0 muß ein Zufallsgenerator beim C 64 eine neue Zahl zwischen 1024 und 1063 erzeugen. VC 20 je nach dem zwischen (7680 oder 4096) + 22 J - Ist k = 1 wird der Gegenstand einfach ein Kästchen nach unten bewegt.

Das ist eigentlich alles, was man für's erste an theoretischem Wissen besitzen muß. Beginnen wir nun mit der BASIC. Dazu ist ein kurzes TBS - Beispielprogramm abgedruckt, das die im Text beschriebenen Funktionen ausführt.

Für den VC 20 gibt es in dieser Ausgabe von HCR den 1. Teil des Listings Galactic-Elster, welches mit Hilfe von TBS erstellt wurde. Mit dem 2. Teil, den Sie in der nächsten Ausgabe finden werden, geben wir Ihnen dann die genaue Spielbeschreibung und versuchen, daran eine weitere Eigenschaft von TBS zu erläutern. ■

Von Mäusen und Fenstern in Computern

In dieser Serie sollen zwei Schlagworte, die die Fachpresse in der letzten Zeit beschäftigt haben, näher erläutert werden und der Hintergrund der damit verbundenen Entwicklung beleuchtet werden. Plastikmäuse und Bildschirmfenster sind das Thema und, so meine ich, wenn über eine wichtige Neuerung berichtet wird, so sollte auch ein neuer Ansatz gefunden werden. Deshalb können Sie das weitere Aussehen dieser Serie entscheidend beeinflussen, indem Sie unter dem Stichwort ,,Fenstermaus'' ihre Fragen und Erfahrungen mit der Maus an mich weiterleiten.

Ein Teil der folgenden Artikel wird immer einem festem Thema gewidmet sein, im zweiten Teil werde ich, soweit möglich, entsprechend auf Ihre Einsendungen reagieren. Die ersten beiden Folgen sollen eine kleine Einführung für Anfänger sein, damit alle später mitreden können.

Von Roderich Bott

Was sind denn nun eigentlich diese Mäuse? Nun, sie sehen alle aus wie ein abgerundeter Quader aus Plastik und haben einen oder mehrere Knöpfe zum Draufdrücken.

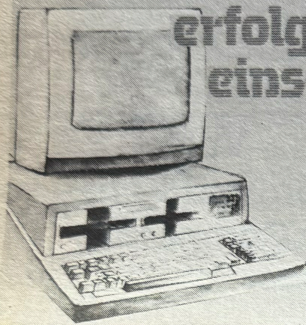
Manche brauchen außerdem noch spezielle Unterlagen, warum, klären wir später.

Da wir jetzt ungefähr wissen wie sie aussehen, können wir nun klären, was sie tun. Auch das ist nicht sehr kompliziert; Sie brauchen sich nur einen Joystick vorzustellen, der den Cursor steuert (für diejenigen, die einen Trackball kennen, ist es noch einfacher: Hier wird das Gehäuse so bewegt, daß die Kugel auf dem Tisch hin und her rollt; eine Art umgekehrter Trackball also).

Die Maus kann das aber besser als der Joystick, weil der Cursor nämlich nicht bei jeder Zeile und jedem Buchstaben, die er überspringt, haltmacht, sondern sie einfach übergleitet, bis man ihn anhält. Damit der Cursor etwas bewirkt, muß noch der (bei mehreren Knöpfen: ein) Knopf gedrückt werden und dann muß die Software die endgültige Position des Cursors festlegen. Dazu braucht man einen recht großen Speicher, da der Cursor jeden Bildschirmpunkt ansprechen kann (Die ganze Sache arbeitet in High Resolution = hochauflösende Grafik). Deshalb hat die Maus auch Schwierigkeiten, sich in 8 bit Rechnern durchzusetzen. Es gibt zwar eine Maus für den Apple II, aber sie hat bei weitem nicht die Qualitäten der Macintosh oder IBM PC Maus. Aber eine neue Generation von Computern wie z. B. der Atari 520 ST (,,Jackintosh'') oder der Amiga von Commodore werden die 16 Bit Prozessoren langsam aber sicher in den Home - Computer - Bereich lotsen und somit die Maus für alle erschwinglich machen. Die Vorteile der Maus liegen klar auf der Hand: In seriösen Anwendungen wie Kalkulation oder Text- / Grafikverarbeitung erleichtert sie die Arbeit erheblich, da der Cursor nun sehr schnell umpositioniert werden kann. Diese Geschwindigkeit ermöglicht Programme, die ganz oder teilweise auf die Tastatur verzichten (z. B. Grafikprogramme) und den Benutzer stattdessen per Mausklick zwischen Alternativen wählen läßt. Dadurch ist er nicht mehr gezwungen, Bücher voll Control Codes auswendigzulernen, sondern kann sich auf die eigentliche Arbeit konzentrieren. Die Nachteile wiegen vor allem dort, wo die Tastatur nach wie vor unumgänglich ist. Hier kann es lästig sein, allzuoft eine Hand von der Tastatur nehmen zu müssen,um die Maus zu steuern. Auch läßt die Positioniergenauigkeit manchmal zu wünschen übrig. Die Maus braucht außerdem in jedem Fall eine glatte Unterfläche. Tischplatten mit einem Furnier werden nach einer Weile aufgerauht und sollten daher durch eine Glas- oder Plastikplatte geschützt werden. Dennoch ist die Maus eine wertvolle Bereicherung des Computerzubehörs.

Die nächste Folge berichtet über Fenster auf der Mattscheibe.... ■

— Bücherecke —



Mikrocomputer erfolgreich einsetzen

von Diplom - Kaufmann Professor Dr. Alfred Moos und Diplom - Ingenieur Professor Pitter A. Steinbuch

1984. 160 Seiten. DM 29,80

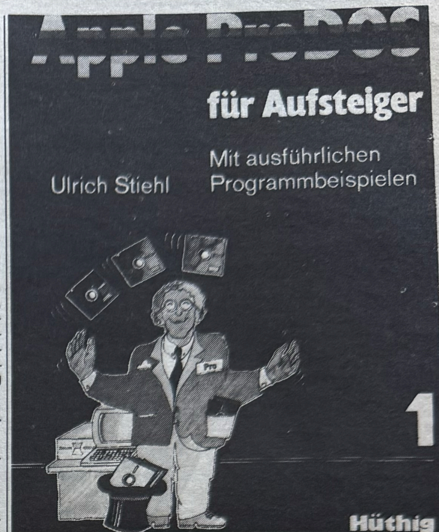
ISBN 3 470 56701 8

Friedrich Kiehl Verlag GmbH Ludwigshafen/Rhein

Mikrocomputer werden seit einiger Zeit in einer Preisklasse angeboten, die auch Klein- und Mittelunternehmen einen wirtschaftlichen Einsatz ermöglichen. Zu klären sind vor der Anschaffung jedoch folgende Fragen:

Bringt ein Mikrocomputer dem Unternehmen Vorteile? Für welche Aufgaben kann er eingesetzt werden? Wie kommt man zu einem optimalen Mikrocomputersystem? Welche EDV - Kenntnisse sind für den Microcomputereinsatz erforderlich?

Die Neuerscheinung „Mikrocomputer erfolgreich einsetzen" beantwortet in allen Details diese Fragen. Checklisten helfen dem Leser bei der Entscheidung für oder wider den Mikrocomputer. Die Checklisten befassen sich u. a. mit den Bereichen Auftragsbearbeitung, Bestellwesen, Buchhaltung, Lagerbestandsrechnung, Lohn- und Gehaltsabrechnung, Rechnungsschreibung, Programmauswahl, Microcomputerarten, Speichergeräte, Drucker, Hardwarevorbereitung, Softwarevorbereitung, Mitarbeiterschulung u. a. Damit steht einem erfolgreichen und problemlosen Einsatz eines Mikrocomputers nichts mehr im Weg. ■



APPLE PRODOS FÜR AUFSTEIGER

Mit ausführlichen Programmbeispielen

von Ulrich Stiehl

Band 1: 1984, 208 S., kart., 28,- DM ISBN 3 - 7785 - 1027 - 4, Dr. Alfred Hüthig Verlag GmbH, 6900 Heidelberg 1

„Apple ProDOS für Aufsteiger" ist der Nachfolgeband zu „Apple DOS 3.3 - Tips und Tricks". Applesoft - Programmierer, die unter DOS 3.3 gearbeitet haben, werden sich schnell an ProDOS gewöhnen, da ProDOS und DOS 3.3 in dieser Hinsicht weitgehend kompatibel sind. Dagegen müssen Assembler - Programmierer völlig umdenken. Deshalb liegt das Schwergewicht dieses Nachfolgebandes auf der Assemblerprogrammierung und der minutiösen Darstellung der ProDOS internen Systemadressen, die jedoch auch für Applesoft - Programmierer von großer Bedeutung sind.

Im ersten Teil wird zunächst ein allgemeiner Überblick über das neue „Professional Disk Operation System" gegeben. Im Anschluß daran folgt eine Gegenüberstellung der Geschwindigkeit des Diskettenzugriffs. Dann wird die interne Speicherorganisation detailliert beschrieben (Boot - Vorgang, Zero - Page, ProDOS - Vektoren, Basic - System - Puffer, Basic - System - Global - Page, Basic - Command - Handler, I/0 - Vektoren, ProDOS - Global - Page, Language - Card - Organisation, Interupt, Disk - Driver, Reboot - Programm usw.) Ebenso ausführlich wird die externe Speicherorganisation geschildert (Spuren, Sektoren, Blocks, Directory - Struktur, Volume Bit Map, Dateistrukturen usw.). Schließlich wird das MLI (Maschine Language Interface) mit zahlreichen praktischen Anwendungsbeispielen erläutert. Insgesamt enthält das ProDOS - Buch ca. 70 Seiten mit eigens für dieses Werk entwickelten Programmen.

— PREISAUSSCHREIBEN —

Auflösung aus Heft Nr. 6/85

Unsere Glücksfee hat aus den vielen richtigen Einsendungen die Gewinner ermittelt.

Die richtige Lösung lautete: **CP/M**

Der erste Preis war ein Sinclair Spectrum Plus

Der 1. Preis geht an: **Dirk Giepen, Schwarzenbruck**
Der 2. Preis geht an: **Ekkehard Uhr, Hennef**
Der 3.-10. Preis geht an: **Helmut Kohlbrecher, Wallenhorst**
Ludger Kappen, Doerpen
Max H. Tobler, CH-Emmenbrücke
Norbert Klein, Siegburg
Michael Frieser, Korntal
Klaus Horst, Konstanz
Torsten Jahn, Oberähren
Anton Floess, Neusaess

Gewinnen Sie einen Cobra-Joystick

Gesucht wird diesmal der Name einer standartisierten Schnittstelle. Diese Schnittstelle überträgt Daten seriell und wird zur DFÜ besonders häufig gebraucht. Den Namen dafür tragen Sie in den Lösungscoupon ein.

Und was gibt es zu gewinnen?

Ein Super Joystick der Marke Cobra ist der 1. Preis.

Hier noch einige Informationen zu dem Cobra-Joystick

Der Cobra - Joystick ist der Joystick schlechthin. Er besitzt eine 1.20 Meter lange Anschlußschnur und drei Feuerknöpfe. Das Innenleben ist beim Cobra - Joystick per Microschalter ausgeführt, was eine sehr hohe Zuverlässigkeit garantiert. Mit Saugnäpfen läßt er sich ohne Probleme auf glatten Flächen befestigen, welches der schon präzisen Funktion zusätzliche Perfektion verleiht. Griffgestaltung und Feuerknopfanordnung sind optimal. Anschließbar ist der Cobra - Joystick an alle Computer mit 9 Pin Eingang (Commodore, Schneider, Atari).

Der Cobra - Joystick wurde von der Fa. Rushware GmbH, An der Gümpgesbrücke, 4044 Kaarst gestiftet.

Der Rechtsweg ist wie immer ausgeschlossen.



— 1. Preis —
Super Joystick Marke Cobra

Gestiftet von Rushware GmbH
An der Gümpgesbrücke, 4044 Kaarst



— 2. Preis —
Softwarepaket
Wert 120.— DM

— 3.-10. Preis —
Softwarepaket
Wert je 40.— DM

— 11.-50. Preis —
1 Gratis Abo HCR

Auflösung des HCR - Heim Computer Report Preisausschreibens:

Gesucht wird diesmal der Name einer standartisierten Schnittstelle. Diese Schnittstelle überträgt Daten seriell und wird zur DFÜ besonders häufig gebraucht.



Name:

Vorname: Alter:

PLZ: Der Rechtsweg ist ausgeschlossen Ort:

- SUPERTEXT -
Schneider CPC 464

Von Frank Thielen und Thomas Barndt

Dokumentation zu Supertext

Dieses Textverarbeitungsprogramm ist bis auf die Bildschirmscrollroutinen nur in Basic geschrieben. Nach dem Start wird als erstes nach der Textdarstellung gefragt. Es besteht die Möglichkeit, mit schwarzer Schrift auf grünem Grund zu schreiben oder umgekehrt.

Danach fragt das Programm nach der maximal einzugebenden Zeilenzahl. Es ist möglich, bis zu 230 Zeilen Text zu schreiben. Schöpft man diese Möglichkeit voll aus, so geht dies jedoch etwas auf Kosten der Geschwindigkeit beim Einfügen oder Löschen von Zeilen. Als nächstes erscheint das Hauptmenue. Dort kann man zwischen folgenden sechs Möglichkeiten wählen:

Eingeben bzw. Edieren von Text
Löschen eines Textes
Ausdrucken eines Textes
Speichern oder Laden von Texten
Ein- oder Ausschalten des deutschen Zeichensatzes
Beenden des Programms.

Beim Laden wird der Inhalt der Cassette auf bereits gespeicherte Texte untersucht. Wird ein Text gefunden, so wird dessen Name, das Aufzeichnungsdatum und die Anzahl der Zeilen angezeigt. Der Benutzer entscheidet dann, ob der Text geladen wird oder nicht.

Das Editieren

Nun zum wichtigsten Punkt, dem Edieren. Nach dem Anwählen dieses Menuepunktes erscheint zunächst der Cursor in der linken oberen Bildschirmecke, sowie die Statuszeile ganz unten. Dort wird die momentane Zeilen- und Spaltenposition alle paar Sekunden angezeigt. Ebenso der momentan eingeschaltete Zeichensatz. Man kann nun seinen Text eingeben.

Über CTRL 'H' (Hilfe) erreicht man die Anzeige der folgenden zur Verfügung stehenden Funktionen:

Rückkehr zum Menue
Springen zu einer gewünschten Zeile
Sichern gelöschter Zeilen
Formatieren des Textes
Einstellen der linken Randbreite
Trennen der Zeile
Vereinigen von Zeilen
Worte suchen
Austauschen von Worten

Zum Formatieren setzt man den Cursor auf die erste Zeile und letzte Spalte des zu formatierenden Textes und betätigt die Tasten CTRL + 'F' (Formatiere) gleichzeitig. Nach Eingabe der Zeilenanzahl werden in diese Zeilen so viele Leerzeichen eingesetzt, bis sie rechtsbündig mit der Spaltenposition abschließen.

Suchen und Austauschen

Das Suchen und Austauschen von Worten geschieht ab Cursorposition abwärts.

Beim Trennen von Zeilen wird die rechte Seite der Zeile, einschließlich der Cursorposition, zwischen die jetzige und die nächste Zeile eingefügt, indem man CTRL + 'T' (Trennen) betätigt. Das Vereinigen von Zeilen erfolgt dementsprechend mit CTRL + 'V'. Wenn der Platz zu diesen Operationen nicht ausreicht, wird dieses in der Statuszeile angezeigt und die Operation nicht durchgeführt. Sollte während des Edierens der Cursor einmal für ein paar Sekunden verschwinden, so ist dies nicht weiter tragisch. Der Computer führt dann eine Sammlung von freiem Speicherplatz durch. Dieses tritt jedoch bei kürzeren Texten sehr selten auf.

Jede gelöschte Zeile wird auf einen Stack (Stapel) gelegt und kann mit CTRL + 'S' (Sichern) wieder angezeigt werden. Man kann auf diese Weise bis zu neun gelöschte Zeilen retten.

Betätigt man CTRL + COPY, so wird die aktuelle Zeile ebenfalls auf den Stack gelegt, jedoch dabei nicht gelöscht. Dadurch ist es möglich, Zeilen an beliebige Textstellen zu kopieren oder zu verschieben.

Retten mit CTRL + S

Das Programm ist an einen Star Drucker DB-510 angepaßt, dessen Steuerzeichen stimmen jedoch weitgehend mit den meisten anderen Druckern überein (z. B. EPSON und kompatible).

Bitte, beachten Sie, daß SUPERTEXT einige kleine Maschinenspracheroutinen enthält. Zeichnen Sie deshalb nach der Eingabe des Programms dieses auf, so daß Sie keinen unerwünschten Absturz erleben. Zum sicheren Funktionieren sollten Sie auch vor jedem Start des Programms den Rechner zurücksetzen.

Noch ein Tip:
Setzen Sie die 'ON ERROR GOTO' - und 'RESUME'-Anweisungen erst zum Schluß in das getestete Programm, so daß Sie eventuelle Fehler auch angezeigt bekommen.

Nachfolgend noch einige Funktionen von Supertext. Bitte, beachten Sie diese ebenso, wie die während des Programmlaufes erscheinenden Anweisungen, damit Ihr Text nicht verloren geht! ■

```
10 MODE 2
20 DEFSTR s-w:DEFINT a-r,x-z
30 KEY DEF 68,1,9,10
40 KEY DEF 9,1,224,223,222
50 KEY DEF 18,1
60 scrollup=""
70 READ byte
80 WHILE byte>=0
90   scrollup=scrollup+CHR$(byte)
100   READ byte
110 WEND
120 szei="engl."
130 INK 0,0:INK 1,26
140 CLS
150 PRINT
160 PRINT
170 PRINT "                              Supertext 1.15"
180 PRINT
190 PRINT
200 PRINT "                         Copyright FTCP + THBCS 1985"
210 PRINT
220 PRINT
230 PRINT "Textdarstellung schwarz auf gruenem Grund (S) oder gruen auf schwar
zem Grund (G)"
240 '
250 '              *** Farben ***
260 '
270 s=UPPER$(INKEY$)
280 IF INSTR(" SG",UPPER$(s))<2 THEN 270
290 IF s="S" THEN BORDER 23:PEN 0:PAPER 1:PEN #1,1:PAPER #1,0:PEN #2,1:PAPER #
2,0 ELSE BORDER 14:PEN 1:PAPER 0:PEN #1,0:PAPER #1,1::PEN #2,0:PAPER #2,1:MID$(s
crollup,4,1)=CHR$(0)
300 DATA &06,&01,&3E,255,&21,0,0,&11,23,79,&CD,&50,&BC,&C9,-1
310 scrolldown=scrollup
320 MID$(scrolldown,2,1)=CHR$(0)
330 scrollinsert=scrolldown:scrolldelete=scrollup
340 CLS
350 WINDOW #1,1,80,25,25    ' Statuszeile
360 WINDOW #2,44,77,8,21    ' Funktionsmenue
370 CLS #1
380 ON ERROR GOTO 390
390 RESUME 400
400 PRINT
410 INPUT "Maximale Zeilenzahl ";maxze
420 IF maxze<24 THEN PRINT:PRINT "Dokument muss laenger als 23 Zeilen sein":GO
TO 400
430 IF maxze>230 THEN PRINT:PRINT "Dokument darf nicht laenger als 230 Zeilen
sein":GOTO 400
440 DIM text(maxze),stack(9) ' Textspeicher + Stack fuer geloeschte Zeilen
450 FOR i=0 TO maxze:text(i)=STRING$(80," "):NEXT
460 FOR i=0 TO 9:stack(i)=STRING$(80," "):NEXT
470 ON ERROR GOTO 480
480 RESUME 490
490 GOSUB 580 ' Menue
500 ON INSTR(" ELACBD",s)-1 GOSUB 750,2990,3100,3550,4050,4370
510 GOTO 490
520 '
530 '     *** Status in Statuszeile (Window #1) ausgeben ***
540 '
550 LOCATE #1,1,1:PRINT#1,USING " Z:###  S:##                    Hilfe : CTRL 'H'
        Zeichensatz: &";ize,isp,szei;
560 RETURN
570 '        ****************************
580 '        **         Hauptmenue     **
590 '        ****************************
600 CLS
610 CLS#1
620 PRINT"E: Eingabe oder Edieren von Text"
630 PRINT"L: Loeschen des Textes"
640 PRINT"A: Ausdrucken des Textes"
650 PRINT"C: Speichern oder Laden von Texten"
660 PRINT"D: Deutscher Zeichensatz"
670 PRINT"B: Beenden des Programms"
680 PRINT
690 PRINT"   Bitte waehlen Sie!"
700 s=UPPER$(INKEY$):IF s="" THEN 700
710 IF INSTR(" ELACBD",s)<2 THEN 700
720 CLS:CLS#1
730 RETURN
740 '
750 '            *** Eingabe bzw. Edieren ***
760 '
770 EVERY 250 GOSUB 530
780 ize=1:isp=1+rand           ' Zeile und Spalte des Cursors im Textarray
790 iba=1                      ' Bildschirmanfang im Textarray (vertikal)
800 icu=1                      ' Cursorposition auf dem Bildschirm (vertikal)
810 GOSUB 2900
820 EI
830 LOCATE isp,icu
840 PRINT CHR$(24);MID$(text(ize),isp,1);CHR$(24);CHR$(8);
850 IF INKEY(23)<>128 THEN s=INKEY$:IF s="" THEN 850 ELSE 880
860 s=INKEY$:IF s="" THEN 860 ELSE IF ASC(s)>31 THEN 880
870 u=CHR$(ASC(s)+96):GOTO 1330
880 IF s<CHR$(32) OR s>CHR$(126) THEN PRINT MID$(text(ize),isp,1);:GOTO 900
890 PRINT s;:MID$(text(ize),isp,1)=s:isp=isp+1:GOTO 2770    ' Druckbares Zeiche
n
900 IF s=CHR$(13) THEN isp=1+rand:ize=ize+1:icu=icu+1:GOTO 2770 ' ENTER
910 IF s=CHR$(242) THEN isp=isp-1:GOTO 2770                    ' Cursor links
920 IF s=CHR$(243) THEN isp=isp+1:GOTO 2770                    ' Cursor rechts
930 IF s=CHR$(240) THEN ize=ize-1:icu=icu-1:GOTO 2770          ' Cursor aufwaerts
940 IF s=CHR$(241) THEN ize=ize+1:icu=icu+1:GOTO 2770          ' Cursor abwaerts
950 IF s=CHR$(224) THEN GOTO 780                               ' COPY (HOME)
960 IF s=CHR$(223) THEN iba=maxze-23:icu=24:ize=maxze:GOSUB 2900:GOTO 830 ' SH
IFT + COPY (Textende)
970 IF s=CHR$(9) THEN isp=isp-isp MOD 10+11:GOTO 2770          ' TAB
980 IF s=CHR$(10) THEN isp=isp-isp MOD 10-9:GOTO 2770          ' SHIFT + TAB
990 IF s<>CHR$(16) AND s<>CHR$(246) THEN 1060
1000 '
1010 '      *** CLR oder SHIFT+Cursor links ***
1020 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+MID$(text(ize),isp+1)+" "
1030 LOCATE 1,icu:PRINT text(ize);:GOTO 2770
1040 '
1050 '        *** DEL ***
1060 IF s<>CHR$(127)THEN 1120
1070 IF isp=1 THEN PRINT#1,TAB(23);CHR$(7);"*** Zeilenanfang erreicht ***":GOTO
830
1080 MID$(text(ize),1)=LEFT$(text(ize),isp-2)+MID$(text(ize),isp)+" "
1090 LOCATE 1,icu:PRINT text(ize);:isp=isp-1:GOTO 2770
1100 '
1110 '      *** SHIFT+Cursor rechts: Zeichen einfuegen ***
1120 IF s<>CHR$(247) THEN 1180
1130 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+" "+MID$(text(ize),isp)
1140 LOCATE 1,icu:PRINT text(ize);:GOTO 2770
1150 '
1160 '        *** SHIFT+Cursor oben: Zeile loeschen ***
1170 '        *** CTRL + COPY: Zeile in Puffer einlesen
1180 IF ABS(ASC(s)-233)<>11 THEN 1250
1190 MID$(stack(ipo),1)=text(ize):ipo=ipo+1:IF ipo>9 THEN ipo=9:FOR i=0 TO 8:MI
D$(stack(i),1)=stack(i+1):NEXT i
1200 IF s=CHR$(222) THEN ize=ize+1:icu=icu+1:GOTO 2770
1210 FOR i=ize TO maxze-1:MID$(text(i),1)=text(i+1):NEXT i:MID$(text(maxze),1)=
STRING$(80," ")
1220 MID$(scrolldelete,6,1)=CHR$(icu-1):CALL PEEK(@scrolldelete+1)+PEEK(@scroll
delete+2)*256:LOCATE 1,24:PRINT text(iba+23);:GOTO 830
1230 '
1240 '        *** SHIFT+Cursor unten: Zeile einfuegen ***
1250 IF s<>CHR$(245) THEN 1310
1260 u=""
1270 IF text(maxze)<>STRING$(80," ") THEN PRINT#1,TAB(25);CHR$(7);"*** Textspei
cher voll ***":GOTO 830
1280 FOR i=maxze TO ize STEP -1:MID$(text(i),1)=text(i-1):NEXT i:MID$(text(ize)
,1)=STRING$(80," ")
1290 MID$(scrollinsert,6,1)=CHR$(icu-1):CALL PEEK(@scrollinsert+1)+PEEK(@scroll
insert+2)*256:IF u="s" THEN 2330 ELSE 830
```

— Funktionen von SUPERTEXT —

SHIFT + Cursor rechts:	Fügt vor dem Cursor ein Zeichen ein.
SHIFT + Cursor links:	Löscht das Zeichen auf Cursorposition.
SHIFT + Cursor unten	Fügt über dem Cursor eine Zeile ein.
SHIFT + Cursor oben	Löscht die Zeile auf Cursorposition.
COPY	Setzt Cursor auf Textanfang.
SHIFT + COPY	Setzt Cursor auf Textende.
CTRL + COPY	Lädt die Zeile in den Puffer.
CTRL + 'S'	Holt gelöschte oder mit CTRL + COPY geladene Zeile und setzt diese zwischen den Cursor und die Zeile darüber.
CLR oder DEL	Diese beiden Tasten besitzen die gewohnten Funktionen.
TAB	Bewegt den Cursor in Zehnerschritten vorwärts
SHIFT + TAB	Bewegt den Cursor in Zehnerschritten rückwärts.

```
1300 '
1310 GOTO 830
1320 '
1330 '             *** Funktionen ***
1340 DI
1350 IF u<>"h" THEN 1550
1360 CLS#2:GOSUB 530
1370 PRINT#2
1380 PRINT#2,"   Ueber 'CTRL' erreichbar:"
1390 PRINT#2
1400 PRINT#2," M: Menue"
1410 PRINT#2," Z: Zeile anwaehlen"
1420 PRINT#2," S: Sichern geloeschter Zeilen"
1430 PRINT#2," F: Formatieren des Textes"
1440 PRINT#2," R: Randbreite einstellen"
1450 PRINT#2," T: Trennen der Zeile"
1460 PRINT#2," V: Vereinigen von Zeilen"
1470 PRINT#2," W: Worte suchen"
1480 PRINT#2," A: Austauschen von Worten"
1490 PRINT#2," H: Anzeigen dieser Information"
1500 LOCATE#1,25,1
1510 PRINT#1,"      < Leertaste! >"
1520 CALL &BB06:GOSUB 2900:GOTO 820
1530 '
1540 '               *** Rueckkehr zum Menue ***
1550 IF u<>"m" THEN 1590
1560 RETURN
1570 '
1580 '              *** Sprung zu Zeile ***
1590 IF u<>"z" THEN 1640
1600 GOSUB 530
1610 LOCATE #1,28,1
1620 INPUT #1," Welche Zeile ";ize:IF ize<1 OR ize>maxze THEN PRINT #1,"Falsche
Zeile. ";:GOTO 1610
1630 iba=MIN(maxze-23,ize):icu=ize-iba+1:GOSUB 2900:GOTO 820
1640 '
1650 '              *** Randbreite einstellen ***
1660 IF u<>"r" THEN 1730
1670 GOSUB 530
1680 LOCATE #1,25,1
1690 INPUT#1," Linke Randbreite ";rand:IF rand<0 OR rand>79 THEN 1690
1700 GOTO 820
1710 '
1720 '              *** Zeile trennen ***
1730 IF u<>"t" THEN 1850
1740 IF text(maxze)<>STRING$(80," ") THEN PRINT#1,TAB(25);CHR$(7);"*** Textspei
cher voll ***":GOTO 820
1750 IF ize=maxze THEN 820
1760 FOR i=maxze TO ize+1 STEP -1:MID$(text(i),1)=text(i-1):NEXT i
1770 MID$(text(ize+1),1)=STRING$(rand," ")+RIGHT$(text(ize),81-isp)+STRING$(MAX
(isp-1-rand,0)," ")
1780 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+STRING$(81-isp," ")
1790 IF icu<23 THEN MID$(scrollinsert,6,1)=CHR$(icu):CALL PEEK(@scrollinsert+1)
+PEEK(@scrollinsert+2)*256
1800 LOCATE 1,icu:PRINTtext(ize);
1810 IF icu<24 THEN LOCATE 1,icu+1:PRINT text(ize+1);
1820 GOTO 820
1830 '
1840 '             *** Zeilen zusammensetzen ***
1850 IF u<>"v" THEN 2030
1860 IF ize=maxze THEN 820
1870 IF text(ize+1)=STRING$(80," ") THEN 2000
1880 il=1
1890 WHILE MID$(text(ize+1),il,1)=" "
1900   il=il+1
1910 WEND
1920 ir=80
1930 WHILE MID$(text(ize+1),ir,1)=" "
1940   ir=ir-1
1950 WEND
1960 IF RIGHT$(text(ize),81-isp)<>STRING$(81-isp," ") THEN PRINT #1:PRINT #1,"
*** Cursor ist nicht am Zeilenende ***";CHR$(7):FOR i=1 TO 3000:NEXT i:GOTO 820
1970 IF LEN(RIGHT$(text(ize),81-isp))<ir-il+1 THEN PRINT #1:PRINT #1,"  *** Pla
tz hinter Cursor nicht ausreichend ***";CHR$(7):FOR i=1 TO 3000:NEXT i:GOTO 820
1980 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+MID$(text(ize+1),il,ir-il+1)
1990 LOCATE 1,icu:PRINT text(ize);
2000 FOR i=ize+1 TO maxze-1:MID$(text(i),1)=text(i+1):NEXT i:MID$(text(maxze),1
)=STRING$(80," ")
2010 MID$(scrolldelete,6,1)=CHR$(icu):CALL PEEK(@scrolldelete+1)+PEEK(@scrollde
lete+2)*256:LOCATE 1,24:PRINT text(iba+23);:GOTO 820
2020 '
2030 '             *** Text Formatieren ***
2040 IF u<>"f" THEN 2290
2050 merk=ize
2060 GOSUB 530
2070 LOCATE#1,25,1
2080 INPUT#1," Wieviele Zeilen ";anze
2090 zeil=ize+anze-1
2100 IF zeil>maxze THEN zeil=maxze
2110 FOR i=ize TO zeil
2120   IF RIGHT$(text(i),80-rand)=STRING$(80-rand," ") THEN 2260
2130   anf=rand+1
2140   WHILE MID$(text(i),anf,1)=" "
2150     anf=anf+1
2160   WEND
2170   du=2

2180   l=INSTR(anf,text(i)," ")
2190   IF RIGHT$(text(i),80-l)=STRING$(80-l," ") THEN 2260
2200   WHILE MID$(text(i),isp,1)=" " AND l<>0
2210     MID$(text(i),1)=LEFT$(text(i),l-1)+" "+MID$(text(i),l)
2220     l=INSTR(l+du,text(i)," ")
2230   WEND
2240   IF MID$(text(i),isp,1)=" " THEN du=du+1:GOTO 2180
2250   IF icu+(i-merk)<25 THEN LOCATE 1,icu+(i-merk):PRINT text(i);
2260 NEXT i
2270 GOTO 820
2280 '
2290 IF u<>"s" THEN 2370
2300 EI
2310 PRINT MID$(text(ize),isp,1);
2320 GOTO 1270
2330 ipo=ipo-1:ipo=MAX(0,ipo)
2340 MID$(text(ize),1)=stack(ipo):LOCATE 1,icu:PRINT text(ize);
2350 GOTO 820
2360 '
2370 '            *** Worte suchen und austauschen ***
2380 IF u<>"w" AND u<>"a" THEN 820
2390 PRINT#1:PRINT#1," Welches Wort suchen Sie ? ";:LINE INPUT#1,such
2400 subst=""
2410 IF u<>"a" THEN 2430
2420 PRINT#1," Mit welchem Wort wollen Sie austauschen ? ";:LINE INPUT#1,subst
2430 t="":diff=LEN(such)-LEN(subst)
2440 ze=ize
2450 PRINT MID$(text(ize),isp,1);
2460 PRINT#1,TAB(30);"Ich suche..."
2470 WHILE ize<maxze+1 AND  t<>"b"
2480   nr=1
2490   WHILE INSTR(nr,text(ize),such)<>0  AND  t<>"b"
2500     ze=ize
2510     icu=ize-iba+1
2520     IF icu>24 THEN iba=MIN(maxze-23,ize):icu=ize-iba+1:GOSUB 2900
2530     isp=INSTR(nr,text(ize),such)
2540     t=""
2550     LOCATE 1,icu:PRINT CHR$(24);text(ize);CHR$(24);:FOR i=1 TO 100:NEXT i:
     LOCATE 1,icu:PRINT text(ize);
2560     LOCATE isp,icu
2570     PRINT CHR$(24);MID$(text(ize),isp,1);CHR$(24);CHR$(8);
2580     IF u="a" THEN PRINT#1,"    W)eitersuchen  A)ustauschen  B)eenden" ELSE
     PRINT#1,"    W)eitersuchen  L)oeschen  B)eenden"
2590     WHILE t="" :t=LOWER$(INKEY$):WEND
2600     PRINT MID$(text(ize),isp,1);
2610     IF t<>"a" AND t<>"l" THEN 2650
2620     IF diff<0 THEN IF RIGHT$(text(ize),-diff)<>STRING$(-diff," ") THEN
     PRINT#1,TAB(23);"*** Platz reicht nicht aus ***";CHR$(7):FOR i=0 TO 15
00:  NEXT:GOTO 2540
2630     MID$(text(ize),1)=LEFT$(text(ize),isp-1)+subst+RIGHT$(text(ize),81-isp
-    LEN(such))+STRING$(ABS(diff)," ")
2640     LOCATE 1,icu:PRINTtext(ize);
2650     nr=isp+1
2660     PRINT#1,TAB(30);"Ich suche..."
2670   WEND
2680   ize=ize+1
2690 WEND
2700 IF t="" THEN PRINT#1,"** Kein '"such"' im Text vorhanden **":GOTO 2730
2710 IF t<>"b" THEN PRINT#1,"** Kein weiteres '"such"' gefunden **":GOTO 2730
2720 GOTO 2740
2730 FOR i=0 TO 3500:NEXT
2740 ize=ze
2750 GOTO 820
2760 '

2770 '              *** Neue Cursorposition ***
2780 '
2790 IF isp=75 THEN PRINTCHR$(7);
2800 IF isp>80 THEN isp=1+rand:ize=ize+1:icu=icu+1
2810 IF isp<1 THEN isp=80:ize=ize-1:icu=icu-1
2820 ' IF isp=80 AND icu=iba+23 THEN isp=79
2830 IF ize=maxze+1 THEN ize=maxze:icu=24:PRINT#1,TAB(22);CHR$(7);"*** Textende
erreicht ***":GOTO 830
2840 IF ize<1 THEN ize=1:icu=1:PRINT#1,TAB(22);CHR$(7);"*** Textanfang erreicht
***":GOTO 830
2850 IF icu>24 THEN iba=iba+1:icu=24:CALL PEEK(@scrollup+1)+PEEK(@scrollup+2)*2
56:LOCATE 1,24:PRINT text(iba+23);:GOTO 830 ' aufwaerts scrollen (neue Zeile ers
cheint unten)
2860 IF icu<1 THEN iba=iba-1:icu=1:CALL PEEK(@scrolldown+1)+PEEK(@scrolldown+2)
*256:LOCATE 1,1:PRINT text(iba);:GOTO 830 ' abwaerts scrollen (neue Zeile ersche
int oben)
2870 GOTO 830
2880 '
2890 '
2900 '       *** Textschirm wiederherstellen ***
2910 '
2920 CLS:CLS#1
2930 FOR i=iba TO MIN(iba+23,maxze)
2940   IF text(i)=STRING$(80," ") THEN PRINT ELSE PRINT text(i);
2950 NEXT i
2960 GOSUB 530
2970 RETURN
2980 '
2990 '          *** Loeschen ***
3000 '
3010 INPUT "Sind Sie sicher, dass Sie den Text loeschen wollen ? (Ja eingeben)
";s
3020 IF s<>"Ja" THEN RETURN
3030 REM Textspeicher mit Leerzeichen fuellen
3040 FOR i=0 TO maxze
3050   MID$(text(i),1)=STRING$(80," ")
3060 NEXT i
3070 MID$(stack(0),1)=STRING$(80," "):ipo=0
3080 RETURN
3090 '
3100 '        *** Ausdrucken ***
3110 '
3120 i=maxze
3130 WHILE text(i)=STRING$(80," ")
3140   i=i-1
3150 WEND
3160 laenge=i
3170 CLS
3180 PRINT "Bitte Drucker vorbereiten und Druckkopf auf Anfang erstes Blatt ste
llen!"
3190 PRINT
3200 PRINT "Automatischer oder manueller Vorschub nach jedem Blatt (A/M) ?"
3210 s=UPPER$(INKEY$)
3220 IF s<>"A" AND s<>"M" THEN 3210
3230 PRINT
3240 CLS
3250 INPUT "Anzahl der Zeichen pro Zeile auf dem Drucker ";druckbreite
3260 IF druckbreite<1 OR druckbreite>255 THEN 3240
3270 WIDTH druckbreite
3280 CLS
3290 INPUT "Anzahl der Kopien ";anzkopie
3300 IF anzkopie<0 THEN 3280
3310 CLS
3320 PRINT "Der Drucker ist nicht angeschlossen oder nicht 'ON LINE' !"
3330 PRINT #8,CHR$(27);"@";
3340 CLS
3350 PRINT #8,CHR$(27);"A";CHR$(15);CHR$(27);"E"; ' Hier spezielle Steuerung fu
er Drucker-Zeilenabstand und Schriftart
3360 druckhoehe=53 ' Anzahl der Zeilen auf einem Blatt, abhaengig vom Zeilenabs
tand
3370 IF szei="deutsch" THEN PRINT #8,CHR$(27);"R";CHR$(2);
3380 FOR j=0 TO anzkopie
3390   FOR i=0 TO laenge
3400     PRINT #8,LEFT$(text(i)+STRING$(175,32),druckbreite);
3410     IF i MOD druckhoehe>0 OR i=0 OR i=laenge THEN 3450
3420     IF s="A" THEN PRINT #8,CHR$(12);:GOTO 3450
3430     INPUT "Naechstes Blatt vorbereiten; wenn fertig, ENTER druecken",s
3440     CLS
3450   NEXT i
3460   PRINT #8
3470   IF j=anzkopie THEN 3510
3480   IF s="A" THEN PRINT #8,CHR$(12);:GOTO 3510
3490   INPUT "Naechstes Blatt vorbereiten; wenn fertig, ENTER druecken",s
3500   CLS
3510 NEXT j

3520 PRINT #8,CHR$(27);"@"; ' wieder initialisieren
3530 RETURN
3540 '
3550 '      *** Speichern bzw. Laden ***
3560 CLS:CLS#1
3570 PRINT"S: Speichern"
3580 PRINT"L: Laden bzw. Textfiles suchen"
3590 PRINT"R: Rueckkehr zum Hauptmenue"
3600 PRINT:PRINT"   Bitte waehlen Sie!"
3610 s=LOWER$(INKEY$)
3620 IF s="r" THEN RETURN
3630 IF s<>"s" THEN 3830
3640 CLS:CLS#1:PRINT#1,TAB(26);"---- TEXT SPEICHERN ----"
3650 IF sdatum="" THEN INPUT "Bitte geben Sie das heutige Datum fuer die Speich
erung an : ",sdatum:PRINT
3660 PRINT"Aufzeichnungsgeschwindigkeit schnell oder langsam (s/l) ?"
3670 t=LOWER$(INKEY$):IF t="" THEN 3670
3680 IF t="s" THEN SPEED WRITE 1 ELSE IF t="l" THEN SPEED WRITE 0 ELSE GOTO 367
0
3690 PRINT:INPUT "Textname : ",sdatei
3700 OPENOUT "supertext.file"
3710 laenge=maxze
3720 WHILE text(laenge)=STRING$(80,32) AND laenge>0
3730   laenge=laenge-1
3740 WEND
```

— Fortsetzung auf Seite 22 —

Biorhythmus

— APPLE II —

Von Oliver Steinmeier

Die Biorhythmik behauptet, daß jeder Mensch ständig drei Zyklen durchläuft, die unterschiedliche Dauer haben. Der physikalische (körperliche) Zyklus hat eine Dauer von 23 Tagen, der emotionale dauert 28 Tage und ein geistiger Zyklus hat eine Länge von 33 Tagen.

Wenn man das Programm startet, muß man zunächst seinen Namen und Geburtsdatum angeben. Dann bestimmt man noch Jahr und Monat, für den der Biorhythmus berechnet wird. Anschließend erfolgt die graphische Darstellung der Bio-Kurven. Sie sind durch entsprechende Buchstaben gekennzeichnet.

Nachdem die Kurve fertiggestellt ist, kommt man an ein Menue. Man kann einen neuen Biorhythmus berechnen, sich die Kurve erneut anschauen, das Programm beenden oder die Graphik ausdrucken. Dabei geht das Programm von einer Druckerkarte in Slot 1 aus.

Außerdem werden Hardcopy - Befehle des Epson - Druckers verwendet, falls man andere Drucker benutzen möchte, sollte man die Befehle in Zeile 550 entsprechend ändern. ■

```
10  REM  ***********************
20  REM  *                     *
30  REM  *                     *
40  REM  *   APPLE-BIORHYTHMUS *
50  REM  *   COPYRIGHT 1984 BY *
60  REM  ***.OLIVER STEINMEIER *

70  REM  PROGRAMMBEGINN

80  GOSUB 870
90  TEXT : HOME : INVERSE : PRINT "  A P P L E  -  B I O R H Y T H M U S
    ": NORMAL
100  PRINT : PRINT  TAB( 4);"COPYRIGHT 1984 BY OLIVER STEINMEIER": POKE 3
    4,6
110  HOME : PRINT : PRINT "BITTE GEBEN SIE IHREN NAMEN AN :": PRINT : INPUT
    "";N$
120  IF  LEN (N$) = 0 THEN 110
130  HOME : PRINT : PRINT "NUN BENOETIGE ICH NOCH IHR GEBURTSDATUM."
140  PRINT "BITTE GEBEN SIE ES IN DER FORM                TT.MM.JJJJ  (Z.B. 25.
    03.1984) AN !"
150  PRINT : INPUT "";G$: IF  LEN (G$) <  > 10 THEN  GOSUB 690: GOTO 150
160 T% =  VAL ( LEFT$ (G$,2)): IF T% > 31 THEN  GOSUB 690: GOTO 150
170 M% =  VAL ( MID$ (G$,4,2)): IF M% > 12 THEN  GOSUB 690: GOTO 150
180 J% =  VAL ( RIGHT$ (G$,4)): IF J% * M% * T% <  = 0 THEN  GOSUB 690: GOTO
    150
190  PRINT : PRINT "FUER WELCHES JAHR SOLL DER BIORHYTHMUS  BERECHNET WER
    DEN (4 STELLEN) ? ": PRINT : INPUT "";JA$
200 JB% =  VAL (JA$): IF JB% < J% THEN  GOSUB 690: GOTO 190
210  PRINT : PRINT "FUER WELCHEN MONAT (1 - 12) ?": PRINT : INPUT "";M$
220 MB% =  VAL (M$): IF MB% < 1 OR MB% > 12 OR (JB% = J% AND MB% < M%) THEN
    GOSUB 690: GOTO 210
230  REM

240  REM   BERECHNUNG DES ALTERS IN TAGEN

250 A = T%:B = M%:C = J%: GOSUB 620:HI = TA
260 A = 1:B = MB%:C = JB%: GOSUB 620:TA = TA - HI
270  HOME : HGR : HCOLOR= 3: SCALE= 2: ROT= 0
280  HPLOT 10,0 TO 10,150: HPLOT 11,75 TO 279,75
290  FOR I = 10 TO 265 STEP 8: HPLOT I,72 TO I,78:H = H + 1: NEXT
300  FOR I = 10 TO 265 STEP 80: HPLOT I,68 TO I,82: NEXT
310  HPLOT 7,150 TO 13,150: HPLOT 7,0 TO 13,0
320  REM

330  REM   KURVEN ZEICHNEN

340 Z% = 23:S = 1: GOSUB 710
350 Z% = 28:S = 3: GOSUB 710
360 Z% = 33:S = 2: GOSUB 710
370  HTAB 1: VTAB 21: PRINT "BIORHYTHMUS FUER ";N$: PRINT "GEBOREN AM ";G
    $
380  VTAB 23: PRINT "ALTER AM 1.";MB%;".";JB%;": ";TA;" TAGE."
390  INPUT "BITTE DRUECKEN SIE <RETURN>";A$
400  TEXT : POKE 34,6: HOME : VTAB 8: PRINT "MENUE:"
410  VTAB 10: PRINT "<1>  GRAPHIK AUSDRUCKEN"
420  VTAB 12: PRINT "<2>  AUF GRAPHIK UMSCHALTEN"
430  VTAB 14: PRINT "<3>  NEUE BERECHNUNG"
440  VTAB 16: PRINT "<4>  ENDE"
```

```
450  VTAB 17: PRINT : PRINT "WAS WOLLEN SIE ? ";: GET Q$
460  IF  VAL (Q$) = 0 OR  VAL (Q$) > 4 THEN 450
470  IF Q$ = "4" THEN  TEXT : HOME : END
480  IF Q$ = "3" THEN  RUN
490  IF Q$ = "2" THEN  POKE  - 16304,0: GOTO 370
500  HOME : VTAB 10: PRINT  TAB( 12);"DRUCKER VORBEREITEN": PRINT  CHR$ (
    4);"PR#1"
510  POKE 1913,65: PRINT  CHR$ (17)
520  PRINT "BIORHYTHMUS FUER ";N$;",": PRINT : PRINT "GEB. AM ";G$
530 : PRINT "ALTER AM 1.";MB%;".";JB%;": ";TA;" TAGE."
540  PRINT : PRINT : PRINT "E R L A E U T E R U N G E N :"
550  PRINT "DIE BIORHYTHMIK BEHAUPTET, DASS ALLE": PRINT "MENSCHEN DREI V
    ERSCHIEDENE ZYKLEN VON": PRINT "UNTERSCHIEDLICHER DAUER DURCHLAUFEN."

560  PRINT : PRINT "ES GIBT DREI ZYKLEN:": PRINT : PRINT "- DER PHYSIKALI
    SCHE (P) DAUERT 23 TAGE"
570  PRINT "- DER EMOTIONALE (E) DAUERT 28 TAGE"
580  PRINT "- DER GEISTIGE ZYKLUS (G) DAUERT 33 TAGE"
590  PRINT : PRINT  CHR$ (4);"PR#0": HOME : GOTO 400
600  END
610  REM

620  REM    BERECHNUNG DER TAGE SEIT DER GEBURT

630  IF M > 2 THEN 660
640 TA = 365 * C + A + 31 * (B - 1) +  INT ((C - 1) / 4) -  INT (3 / 4 *
    INT ((C - 1) / 100 + 1))
650  RETURN
660 TA = 365 * C + A + 31 * (B - 1) -  INT (.4 * B + 2.3) +  INT (C / 4) -
    INT (3 / 4 *  INT (C / 100 + 1))
670  RETURN
680  REM  FALSCHE EINGABE

690  PRINT : PRINT  CHR$ (7);"FALSCHE EINGABE !": RETURN
700  REM

710  REM  PLOT-UNTERROUTINE

720 I = TA: GOSUB 820
730  DRAW S AT 4, - Y * 75 + 79
740  HPLOT 10, - Y * 75 + 75
750  FOR I = TA + 1 TO TA + 31
760  GOSUB 820
770  HPLOT  TO (I - TA) * 8 + 10, - Y * 75 + 75
780  NEXT I
790  DRAW S AT 270, - Y * 75 + 79
800  RETURN
810  REM

820  REM   DATEN-BERECHNUNG

830 BE = I * 360 / Z%
840 Y =  SIN (BE * .0174533)
850  RETURN
860  REM

870  REM    POKE SHAPE IN SPEICHER

880  FOR I = 768 TO 796: READ A: POKE I,A: NEXT
890  POKE 232,0: POKE 233,3
900  RETURN
910  REM

920  REM   SHAPE-DATEN

930  DATA 3,0,8,0,13,0,20,0,36,36,45,54,63,0,63,54,54,45,36,39,0,63,54,45
    ,10,19,63,36,0
```

Fortsetzung aus Heft 7/85

Mathe 1 V 16 C

Von Uwe Haferland

2.) Monotonieverhalten von Funktionen

Hier wird eine Funktion vorher wie eben auf Grenzwerte untersucht, bevor die eigentliche Berechnung folgt. So wird erst einmal festgestellt, ob die Funktion überhaupt lösbar ist, wobei bei Lösbarkeit die Funktion an einem beliebigen Punkt auf a) streng monoton steigend b) streng monoton fallend und c) stagnierend untersucht wird. Bei Nichtlösbarkeit weist das Programm natürlich auf den Grund hin.

3.) Integralrechnung

Bei Berechnung von Integralen ist normalerweise eine Nullstellenbestimmung notwendig, um negative Flächen getrennt zu behandeln. Diese mühevolle Prozedur entfällt bei diesem Programm, da eine Softwareroutine negative Flächen erkennt und entsprechend behandelt.

4.) Nullstellenberechnung von Funktionen beliebigen Grades.

Hier wird die Newtonsche Näherung angewendet, da diese bei extrem kurzer Rechenzeit hochpräzise Ergebnisse liefert. Auch hier liegt eine hohe Genauigkeit des Ergebnisses vor, typisch sind vier Stellen hinter dem Komma. Sollten bei irrationalen Funktionen infolge Definitionslücken Nullstellen im Zähler und im Nenner auftreten, so prüft das Programm, ob überhaupt noch eine Lösung existieren kann und gibt bei Nichtlösbarkeit auch den Grund an. Gerade diese absolute Immunität gegenüber Unendlichkeitsstellen dürfte wohl selten bei Mathematik-Programmen der Fall sein, von denen es überhaupt sehr wenige gibt. Nullstellenberechnung ist Schülern der unteren Klassen unter dem Namen "Lösungen von quadratischen und kubischen Gleichungen" bekannt, wobei alle Grade innerhalb der Rechnerkapazität zugelassen sind.

5.) Dualzahlen/Dezimalzahle

Hier können bis zu 600stellige Dualzahlen eingegeben werden, wobei am Ende der Eingabe die entsprechende Dezimalzahl erscheint. Bei beiden Dualzahlroutinen können auch kommabehaftete Zahlen (Gleitkommaarithmetik) verarbeitet werden.

6.) Dezimalzahlen/Dualzahlen

Man gibt eine Dezimalzahl ein, wobei die entstehende Dualzahl bis zu 600 Stellen lang sein darf. Die Stellenanzahl wird angezeigt, sodaß ein lästiges Zählen entfällt.
Für Assembler- und Maschinenspracheprogrammierer sind diese beiden Unterprogramme unentbehrlich.

7.) Determinanten

Hiermit können Gleichungen mit bis zu vier Unbekannten gelöst werden. Auf nicht lösbare Gleichungen wird hingewiesen. Alle Rechenarten werden komfortablerweise über Menukarte angesteuert, erfordern aber den 48 Kb Spectrum. Da das Programm modular aufgebaut ist, können einzelne Rechenarten ihm entnommen und auf andere Rechner umgeschrieben werden. Alle Rechenarten funktionieren unabhängig voneinander.

Sonstiges

Bedenken Sie, daß der Rechner durch eine weniger gute Romroutine nicht in der Lage ist, (x)↑n zu berechnen, wenn x negativ und n beliebig ist. Hier treten Ungereimtheiten auf. Um dies zu umgehen, betrachtet man folgendes Beispiel: Y=X↑3, X soll -2 sein. Man gibt ein: X*X*X anstatt X↑3. Auch bei positiven Argumenten (X) sollte man dieses Verfahren wählen, wie Erfahrungen zeigen. Der Rechner arbeitet bei trigonometrischen Funktionen wie sin X, tan X, etc. in Bogenmaß. Soll er in Grad arbeiten, so ist folgende Eingabe notwendig: Beispiel Y=sin X, in Grad muß es heißen: sin (X*Pi/180). Der Klammerausdruck bleibt bei allen trigonometrischen Funktionen bestehen. Benutzen Sie nie die Break- oder Stoptasten, da das Programm sonst abstürzt. Wollen Sie trotzdem Eingriffe im Programm vornehmen, so lassen Sie eine Funktion zu Ende zeichnen, und betätigen Sie dann zwei Mal die Taste "U". Da beim Spectrum bedauerlicherweise hyperbolische Funktionen völlig fehlen, kann man sie mit Hilfe folgender gleichwertiger Ersatzfunktionen trotzdem berechnen:
cosh X=(exp x+exp (-x))/2
tanh X=(exp (2*x) - 1//exp(2*x) + 1); sinh X = (exp X-exp (-x))/2
coth X = (exp (2*x) + 1) / (exp (2*x) -1)
Bedenken Sie bitte, daß bei vielen Rechenarten nur Näherungsverfahren angewendet werden können, so daß Fehler von max. 0,01%auftreten können, die aber völlig vernachlässigt werden dürfen.
Achtung: Beim Spectrum Issue 1 und Issue 2 kann es infolge einer fehlerhaften Romroutine bei der Umwandlung von kommabehafteten Dezimalzahlen >800 in Dualzahlen passieren, daß der Rechner sich um mehrere zehntausendstel verrechnet, und somit mehr Stellen hinter dem Komma erzeugt als erwartet. Bei natürlichen Zahlen konnte dieser Fehler softwaremäßig beseitigt werden. ■

Darstellung zum Artikel

```
 520 REM INTEGRALE
 521 LET K=0.5
 522 LET FEHLER=1
 525 CLS : PRINT AT 0,13;"INTEGRALE"
 530 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABH
ENGIGE VARIABLE Y.:";A$
 535 INPUT "UNTERE GRENZE:",D,"OBERE GRENZE:",E
 536 IF E-D>=10 THEN  LET K=1.5
 537 PRINT AT 2,0;"WAEHLEN SIE CA.";INT ((E-D)^0.25*K*20);"INTERVALLE."
 538 PRINT "BENUTZEN SIE NUR GERADE ZAHLEN!"
 542 INPUT "SUBINTERVALLANZAHL:";F
 543 CLS
 545 IF F>=12 THEN  PRINT "BITTE WARTEN!"
 550 LET B=(E-D)/2/F
 555 LET A=0: LET X=D: GO SUB 605
 560 FOR G=1 TO F
 565 LET A=Y+A: LET X=X+B: GO SUB 605
 570 LET A=Y*4+A: LET X=X+B: GO SUB 605
 575 LET A=Y+A
 580 NEXT G
 585 LET C=A*B/3
 595 BEEP .5,30 : BEEP 0.5,27
 600 PRINT "DIE FLAECHE BETRAEGT",(C*100)/100
 602 GO TO 815
 610 LET Y=ABS VAL A$
 615 RETURN
 655 REM NEWTONSCHE NAEHERUNG
 660 LET FEHLER=2
 665 CLS : PRINT AT 0,0;"NULLSTELLENBESTIMMUNG MIT       NEWTONSCHER NAEHERUNG
 "
 670 INPUT "GEBEN SIE NUN DIE FUNKTION EIN  MIT X ALS ARGUMENT,ABER OHNE DIEABH
ENGIGE VARIABLE Y.HANDELT ESSICH UM EINE GEBROCHEN RATIONALEFUNKTION,SO GEBEN S
E NUR DEN   ZAEHLER EIN:";A$
 671 INPUT "GEBEN SIE DEN NENNER EIN,WENN ESSICH UM EINE GEBROCHEN RATIONALEFUN
TION HANDELN SOLLTE,ANDERN- FALLS GEBEN SIE 1 EIN:";B$
 675 LET M=0: DIM Y(3): OVER 0
 680  LET B=0: LET F=999999999: LET A=0.00001
 685 INPUT "GEBEN SIE EINE GESCHAETZTE      NULLSTELLE EIN.VERMUTEN SIE     WEI
ERE,SO STARTEN SIE DIESES   PROGRAMM ERNEUT,ALLERDINGS MIT  EINEM ANDEREN SCHAE
ZWERT:",E
 690 LET N=1E-5
 695 LET X=VAL "E"
 696 LET Z=E
 700 LET M=M+1
 705 IF M=100 THEN  PRINT AT 19,0;"KEINE NULLSTELLE GEFUNDEN!      ": PAUSE 450
 GO TO 110
 710 PRINT AT 8,0;"NAEHERUNG NR.";M
 715  LET B=1
 720  GO SUB 800
 725  LET X=X+A
 730 LET B=2
 735  GO SUB 800
 740 LET B=3
 745  LET X=X-2*A
 750  GO SUB 800
 775 IF ABS (F-E)>N THEN  GO TO 695
 780 IF ABS (F-E)<=N THEN  GO TO 810
 790 LET F=VAL "E"
 800 LET Y(B)=VAL A$
 805  RETURN
 810 LET X=E
 811 IF ABS VAL B$<=1E-4 THEN  PRINT AT 12,0;"AN DIESER STELLE IST DER NENNER DE
R FUNKTION EBENFALLS NULL,SO  DASS HIER WAHRSCHEINLICH KEINE  NULLSTELLE VORLIE
T.GRENZWERT-  UNTERSUCHUNGEN MUESSTEN VORGE-  NOMMEN WERDEN,UM DIE FRAGE ZU   KL
AEREN,OB DAS ERGEBNIS EINE    NULLSTELLE UEBERHAUPT DARSTELLENKANN."
 812 IF ABS VAL B$<=1E-4 THEN  GO TO 815
 813 IF ABS VAL B$>1E-4 THEN  PRINT AT 19,0;"DIE NULLSTELLE LIEGT BEI X=";INT (
E*1E5)/(-1E5);"    "
 814 IF ABS VAL B$<=1E-4 THEN  PRINT "EVENTUELLE NULLSTELLE:";INT (-E*1E4)/(-1E
)
```

Darstellung zum Artikel

```
 900 REM DUALZAHLEN
 901 CLS : LET WARNING=0
 902 LET KARIN=0: LET KORREKTUR=0
 907 PRINT "DRUECKEN SIE 1 FUER DIE UMWAND- LUNG VON DEZIMALZAHLEN IN DUAL- ZAHL
EN,ODER 2 FUER DUALZAHLEN INDEZIMALZAHLEN:"
 908 IF INKEY$="2" THEN  GO TO 1000
 909 IF INKEY$="1" THEN  GO TO 911
 910 GO TO 908
 911 LET D=-1: LET C=2
 912 INPUT "GEBEN SIE EINE POSITIVE,REELLE  DEZIMALZAHL EIN:";A
 913 LET B=ABS INT (-LN (A+1)/LN 2): IF A-INT A=0 THEN  LET KORREKTUR=1
 914 CLS : IF A=0 THEN  PRINT "  DIE DUALZAHL LAUTET:0"
 915 LET BN=B
 916 IF A=0 THEN  GO TO 815
 917 LET B=B-1
 918 LET D=D+1
 920 IF D=31 THEN  LET C=C+1
 922 IF D=31 THEN  LET D=0
 924 IF B=9 THEN  LET A=A-2^9: GO TO 927
 925 LET A=A-2^B
 930 IF A<0 THEN  GO TO 960
 935 PRINT AT C,D;1
 936 IF A>=-4E-5 AND A<=4E-5 AND KARIN=1 THEN  GO TO 980
 940 IF (A+B)=0 THEN  GO TO 980
 941 IF KARIN=1 AND A=0 THEN  GO TO 980
 942 IF B=0 AND A<>0 THEN  GO TO 971
 945 GO TO 917
 960 LET A=A+2^B
 962 IF BN-B=1 THEN  LET WARNING=1
 965 IF BN-B<>1 THEN  PRINT AT C,D;0
 967 IF (A+B)=0 THEN  GO TO 980
 968 IF KARIN=1 AND A=0 THEN  GO TO 980
 969 IF B=0 AND A<>0 THEN  GO TO 971
 970 GO TO 917
 971 LET D=D+1
 972 PRINT AT C,D;","
 973 LET KARIN=1
 974 GO TO 917
 980 PRINT AT 0,0;"DIE ";BN-B-WARNING;"STELLIGE DUALZAHL LAUTET:"
 981 GO TO 815
 990 REM  IN  DEZIMALZAHLEN
1000 CLS : PRINT "         BEISPIEL:              1000,001__BIT-3            !  BIT0
   !  !  !                 !  ! BIT-1
               BIT3"
1010 INPUT "ANZAHL DER STELLEN VOR DEM      KOMMA:";H
1020 INPUT "ANZAHL DER STELLEN NACH DEM     KOMMA:";NACH
1025 LET K=0: LET J=-1-NACH
1030 CLS : LET J=J+1: PAUSE 20: PRINT "BIT";J;":"
1031 IF J=H THEN  GO TO 1060
1040 IF INKEY$="1" THEN  GO TO 1070
1041 IF INKEY$="0" THEN  GO TO 1030
1045 GO TO 1040
1050 NEXT J
1060 CLS : PRINT "DIE DEZIMALZAHL LAUTET:         ";K: GO TO 815
1070 LET K=K+2^J
1071 CLS
1072 PRINT "BIT";J: GO TO 1030
1100 REM DETERMINANTN
1101 CLS
1105 PRINT AT 0,6;"DETERMINANTEN"
1110 INPUT "ANZAHL DER UNBEKANNTEN:";an
1112 DIM a(14): DIM b(14): DIM c(14): DIM d(14): DIM e(6): DIM  v(6)
1115 IF an=2 THEN  GO TO 1200
1120 IF an=3 THEN  GO TO 1250
1125 IF an=4 THEN  GO TO 1321
1200 PRINT "DIE GLEICHUNGEN HABEN FOLGENDE  FORM: a(1)*x+b(1)*y=c(1)
 a(2)*x+b(2)*y=c(2)"
1205 PRINT "GEBEN SIE DIE KOEFFIZIENTEN IN  DER ABGEFRAGTEN REIHENFOLGE EIN:"
```

```
1210 INPUT "a(1):";a(1);TAB 32;"b(1):";b(1);TAB 32;"c(1):";c(1);TAB 32;"a(2):";
(2);TAB 32;"b(2):";b(2);TAB 32;"c(2):";c(2)
1215 LET D=a(1)*b(2)-b(1)*a(2)
1220 IF D=0 THEN  PRINT "DIE GLEICHUNGEN SIND NICHT      LOESBAR!": GO TO 815
1225 LET DX=c(1)*b(2)-c(2)*b(1)
1230 LET DY=a(1)*c(2)-c(1)*a(2)
1234 CLS : PRINT "HIER DIE ERGEBNISSE:"
1235 PRINT "X=";DX/D;"    Y=";DY/D
1237 PRINT AT 8,0;"KOEFFIZIENTENDETERMINANTE=";D
1240 GO TO 815
1250 PRINT "DIE GLEICHUNGEN HABEN FOLGENDE  FORM:  a(1)*x+b(1)*y+c(1)*z=d(1)
 a(2)*x+b(2)*y+c(2)*z=d(2)        a(3)*x+b(3)*y+c(3)*z=d(3)"
1255 PRINT AT 7,0;"GEBEN SIE DIE KOEFFIZIENTEN IN  DER ABGEFRAGTEN REIHENFOLGE
IN:"
1260 INPUT "a(1):";a(1);TAB 32;"b(1):";b(1);TAB 32;"c(1):";c(1);TAB 32;"d(1):";
(1);TAB 32;"a(2):";a(2);TAB 32;"b(2):";b(2);TAB 32;"c(2):";c(2);TAB 32;"d(2):";
(2);TAB 32;"a(3):";a(3);TAB 32;"b(3):";b(3);TAB 32;"c(3):";c(3);TAB 32;"d(3):";
(3)
1265 LET DKOEF=a(1)*(b(2)*c(3)-c(2)*b(3))-a(2)*(b(1)*c(3)-b(3)*c(1))+a(3)*(b(1)*
c(2)-c(1)*b(2))
1266 IF DKOEF=0 THEN  PRINT AT 16,0;"DIE GLEICHUNGEN SIND NICHT      LOESBAR!":
GO TO 815
1270 LET DX=d(1)*(b(2)*c(3)-c(2)*b(3))-d(2)*(b(1)*c(3)-b(3)*c(1))+d(3)*(b(1)*c(
)-c(1)*b(2))
1275 LET DY=a(1)*(d(2)*c(3)-c(2)*d(3))-a(2)*(d(1)*c(3)-d(3)*c(1))+a(3)*(d(1)*c(
)-c(1)*d(2))
1280 LET DZ=a(1)*(b(2)*d(3)-d(2)*b(3))-a(2)*(b(1)*d(3)-b(3)*d(1))+a(3)*(b(1)*d(
)-d(1)*b(2))
1285 LET x=DX/DKOEF: LET y=DY/DKOEF: LET z=DZ/DKOEF
1290 CLS : PRINT "HIER DIE ERGEBNISSE:"
1295 PRINT "x=";x;"  y=";y;"  z=";z
1300 PRINT AT 8,0;"KOEFFIZIENTENDETERMINANTE=";DKOEF
1320 GO TO 815
1340 PRINT "DIE GLEICHUNGEN HABEN FOLGENDE  FORM:                          a(1
*u+b(1)*x+c(1)*y+d(1)*z=e(1)a(2)*u+b(2)*x+c(2)*y+d(2)*z=e(2)a(3)*u+b(3)*x+c(3)*
+d(3)*z=e(3)a(4)*u+b(4)*x+c(4)*ykd(4)*z=e(4)"
1345 RESTORE 1420
1350 PRINT AT 7,0;"GEBEN SIE DIE KOEFFIZIENTEN IN  DER ABGEFRAGTEN REIHENFOLGE
IN:"
1360 INPUT "a(1):";a(1);TAB 32;"b(1):";b(1);TAB 32;"c(1):";c(1);TAB 32;"d(1):";
(1); TAB 32;"e(1):";e(1);TAB 32;"a(2):";a(2);TAB 32;"b(2):";b(2);TAB 32;"c(2):"
c(2);TAB 32;"d(2):";d(2);TAB 32;"e(2):";e(2);TAB 32;"a(3):";a(3);TAB 32;"b(3):"
b(3);TAB 32;"c(3):";c(3);TAB 32;"d(3):";d(3);TAB 32;"e(3):";e(3);TAB 32;"a(4):"
a(4);TAB 32;"b(4):";b(4);TAB 32;"c(4):";c(4);TAB 32;"d(4):";d(4);TAB 32;"e(4):"
e(4)
1410 READ a(11),a(12),a(13),a(14),b(11),b(12),b(13),b(14),c(11),c(12),c(13),c(1
),d(11),d(12),d(13),d(14),n
1420 DATA a(1),a(2),a(3),a(4),b(1),b(2),b(3),b(4),c(1),c(2),c(3),c(4),d(1),d(2)
d(3),d(4),1
1425 GO SUB 1600
1430 READ a(1),a(2),a(3),a(4),n
1435 DATA e(1),e(2),e(3),e(4),2
1440 GO SUB 1600
1445 READ a(1),a(2),a(3),a(4)
1450 DATA a(11),a(12),a(13),a(14)
1452 CLS
1455 READ b(1),b(2),b(3),b(4),n
1460 DATA e(1),e(2),e(3),e(4),3
1465 GO SUB 1600
1470 READ b(1),b(2),b(3),b(4)
1475 DATA b(11),b(12),b(13),b(14)
1480 READ c(1),c(2),c(3),c(4),n
1485 DATA e(1),e(2),e(3),e(4),4
1490 GO SUB 1600
1495 READ d(1),d(2),d(3),d(4),c(1),c(2),c(3),c(4),n
1500 DATA e(1),e(2),e(3),e(4),c(11),c(12),c(13),c(14),5
1505 GO SUB 1600
1510 IF v(1)=0 THEN  PRINT "DIE GLEICHUNGEN SIND NICHT      LOESBAR!": GO TO 815
1520 LET u=v(2)/v(1): LET x=v(3)/v(1): LET y=v(4)/v(1): LET z=v(5)/v(1)
1525 PRINT "HIER DIE ERGEBNISSE:                       u=";u;" x=";x;" y=";y;" z=";z
1530 PRINT AT 8,0;"KOEFFIZIENTENDETERMINANTE=";v(1)
1598 GO TO 815
1600 LET v(n)=a(1)*(b(2)*(c(3)*d(4)-c(4)*d(3))-b(3)*(c(2)*d(4)-c(4)*d(2))+b(4)*(
c(2)*d(3)-d(2)*c(3)))-a(2)*(b(1)*(c(3)*d(4)-d(3)*c(4))-b(3)*(c(1)*d(4)-d(1)*c(4
)+b(4)*(c(1)*d(3)-d(1)*c(3)))+a(3)*(b(1)*(c(2)*d(4)-d(2)*c(4))-b(2)*(c(1)*d(4)-c
(4)*d(1))+b(4)*(c(1)*d(2)-c(2)*d(1)))-a(4)*(b(1)*(c(2)*d(3)-c(3)*d(2))-b(2)*(c(1
)*d(3)-c(3)*d(1))+b(3)*(c(1)*d(2)-c(2)*d(1)))
1605 RETURN
9900 IF FEHLER=1 THEN  PRINT "DIE FUNKTION IST AN DIESER      STELLE UNSTETIG UN
D DAHER NICHT LOESBAR!DRUECKEN SIE BITTE EINE TASTE.": GO TO 815
9910 IF FEHLER=3 THEN  LET X=X+0.1: GO TO 695
9920 IF X<B THEN  LET X=X+C
9921 IF INKEY$="U" THEN  PRINT USR 63584: STOP
9922 GO TO 440
```

Hilbert- und Sierpinski-Kurven in SuperBASIC mit dem QL

Das SuperBASIC des Sinclair QL ist wirklich super!

Der Befehlssatz ist nicht nur umfangreich, er gestattet auch mit seinen PASCAL-Elementen und seinen LOGO-Befehlen "strukturiertes Programmieren".

Als Beispiel hierzu diene die Rekursion, die anhand zweier mathematisch interessanter Kurven in den beiden folgenden Programmen dargestellt sein soll.

0 Allgemeines zur Rekursion

" Ein Objekt heißt »rekursiv«, wenn es sich selbst als Teil enthält oder mit Hilfe von sich selbst definiert ist" (Nikolaus Wirth, PASCAL- und MODULA-2- Erfinder).

Steht man vor dem Problem, eine unendliche Menge von Ob- jekten durch eine unendliche Aussage zu beschreiben, oder eine unendliche Anzahl Berech- nungen ohne explizit definierte Schleifen durchzuführen, so ist ein endliches rekursives Pro- gramm notwendig.

Mit Hilfe der Prozedur (PROCedure) gelingt es in SuperBASIC, elegante, rekursive Algorithmen zu schreiben.

Ruft eine Prozedur P sich explizit selbst auf, so heißt P direkt rekursiv. Enthällt P eine Prozedure Q, die P direkt oder indirekt aufruft, so heißt P indirekt rekursiv.

1. Hilbert Kurven

Fig. 1 besteht aus 5 übereinander gezeichneten "Kurven", denen ein sich wandelndes, regelmäßiges Muster zugrundeliegt. Nach welchem Algorithmus (Rekursionsschema) ist ein solches Muster entstanden?

Die Figuren 2 - 4 zeigen diese Überlagerungskurven voneinander getrennt gezeichnet. Der Mathematiker kennzeichnet sie mit H (1), H (2), H (3),...,H(n); *H(n) heißt Hilbert Kurve n-ter Ordnung,* nach ihrem Entdecker David Hilbert (1891).

Betrachten wir die einzelnen Kurven genauer, so stellen wir fest: jede Kurve H(n) setzt sich aus 4 Einheiten der Kurve H(n-1) zusammen. Die Einheitslänge wird halbiert, dann wird geeignet rotiert und schließlich mit 3 Geraden verbunden. Dabei ist H(1) aus 4 Einheiten der leeren Figur H(0) entstanden;hier wurden nur die 3 Verbindungsstrecken gezeichnet. Nun, jede der Kurven H(n) besteht aus 4 halbgroßen Kopien von H(n-1), und eine Prozedure, die H(n) zeichnen kann, besteht aus 4 Stücken, von denen .jedes (H(n-1) in richtiger Größe und in einem richtigen Winkel gezeichnet wird. Im Programm 1 heißen die 4 Stücke (Prozeduren) a, b, c, d, und die Verbindungsroutinen TURNTO I : MOVE h.

Das Rekursionsschema hat dann folgende Gestalt:

a: d←a↓a→b
b: c↑b→b↓a
c: b→c↑c←d
d: a↓d←d↑c

Die Prozedur a ruft also rekursiv die Prozeduren d,a,a,b nacheinander auf. Für jede zu überlagernde Hilbert-Kurve wird a vom Hauptprogramm abgerufen. Im Hauptprogramm selbst werden, mit h als Einheitsmaß, die Startwerte der Koordinaten x,y berechnet. Die Größe hØ hat die Gestalt hØ = 2^k, ist also eine Zweierpotenz und k≧n. Im Programm ist k=6, denn $2^6$6=64.

Zusammenfassung:
Zeilen 15o - 22o Initialisierung
Zeilen 26o - 32o Zeichnen der Hilbert-Kurve i-ter Ordnung
Zeilen 34o - 7oo Rekursive Prozeduren a(i), b(i), c(i), d(i)

```
100 REMark H I L B E R T - K U R V E N
110 :
120 REMark (C) 1985  Harald BENSOM
130 :
140 :
150 REMark Initialisierung
160 :
170 WINDOW 512,256,0,0
180 MODE 0:PAPER 0:INK 7:CLS
190 PENDOWN:TURNTO 0
200 n=5:h0=64
210 i=0:h=h0
220 x=(h DIV 2)+h:y=h DIV 2
230 :
240 REMark Hilbert-Kurve der Ordnung i
250 :
260 REPeat loop
270  i=i+1:h=h DIV 2
280  x=x+(h DIV 2):y=y+(h DIV 2)
290  POINT x-30,y+20
300   a i
310  IF i=n THEN EXIT loop
320 END REPeat loop
330 :
340 REMark Rekursive Prozeduren
350 :
360 DEFine PROCedure a(i)
370  IF i>0 THEN
380   d i-1:TURNTO 180:MOVE h
390   a i-1:TURNTO 270:MOVE h
400   a i-1:TURNTO 0:MOVE h
410   b i-1
420  END IF
430 END DEFine a
440 :
450 DEFine PROCedure b(i)
460  IF i>0 THEN
470   c i-1:TURNTO 90:MOVE h
480   b i-1:TURNTO 0:MOVE h
490   b i-1:TURNTO 270:MOVE h
500   a i-1
510  END IF
520 END DEFine b
530 :
540 DEFine PROCedure c(i)
550  IF i>0 THEN
560   b i-1:TURNTO 0:MOVE h
570   c i-1:TURNTO 90:MOVE h
580   c i-1:TURNTO 180:MOVE h
590   d i-1
600  END IF
610 END DEFine c
620 :
630 DEFine PROCedure d(i)
640  IF i>0 THEN
650   a i-1:TURNTO 270:MOVE h
660   d i-1:TURNTO 180:MOVE h
670   d i-1:TURNTO 90:MOVE h
680   c i-1
690  END IF
700 END DEFine d
```

Fig.2-4: Hilbert-Kurven H(1),H(2),H(3) (getrennt gezeichnet)

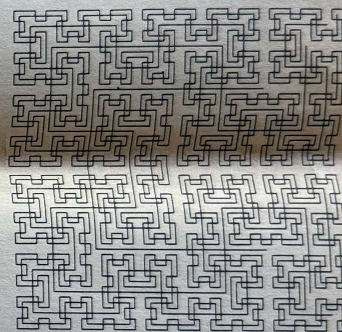


Fig.1: Hilbert-Kurven H(1),...,H(5) (übereinander gezeichnet)

2. Sierpinski Kurven

Das Muster in Fig. 5 zeigt eine Überlagerung von 4 komplexeren Kurven, den sog. Siepinski Kurven (Polnischer Mathematiker). Bezeichnen wir diese mit S(1), S(2), S(3), S(4),...,S(n) (vgl. die Fig. 6 und 7).

Der wesentliche Unterschied zu den Hilbert-Kurven ist ihre Geschlossenheit. Und damit haben wir auch schon den Algorithmus. Die Basiskurve muß offen sein, und die Verbindung darf nicht zur Rekursion selbst gehören! Diese 4 Verbindungsstrecken befinden sich in den äußeren 4 Eckstücken.

Das Rekursionsschema sieht dann folgendermaßen aus:

a: a↘b→d↗a
b: b↗c↓a↘b
c: c↖d←b↙c
d: d↗a↑c↖d

Die waagerechten und senkrechten Pfeile kennzeichnen Strecken doppelter Länge.

Das Basismuster hat die Gestalt:
Basis: a↘b↙c↖d↗

Dieser Algorithmus steht im Hauptprogramm und wird nacheinander aufgerufen.

Zusammenfassung:
Zeilen: 15o-22o: Initialisierung
Zeilen 26o-35o:Zeichnen der Siepinski-Kurve i-ter Ordnung
Zeilen 37o-73o Rekursive Prozeduren a(i),...,d(i).

Darstellung zum Artikel

```
100 REMark S I E R P I N S K I - K U R V E N
110 :
120 REMark (C) 1985  Harald BENSOM
130 :
140 :
150 REMark Initialisierung
160 :
170 WINDOW 512,256,0,0
180 MODE 0:PAPER 0:INK 7:CLS
190 PENDOWN:TURNTO 0
200 n=4:h0=64
210 i=0:h=h0 DIV 4
220 x=2*h:y=3*h
230 :
240 REMark Sierpinski-Kurve der Ordnung i
250 :
260 REPeat loop
270   i=i+1:x=x-h
280    h=h DIV 2:y=y+h
290    POINT x+40,y+20
300    a i:TURNTO 315:MOVE h
310    b i:TURNTO 225:MOVE h
320    c i:TURNTO 135:MOVE h
330    d i:TURNTO 45:MOVE h
340   IF i=n THEN EXIT loop
350 END REPeat loop
360 :
370 REMark Rekursive Prozeduren
380 :
390 DEFine PROCedure a(i)
400   IF i>0 THEN
410     a i-1:TURNTO 315:MOVE h
420     b i-1:TURNTO 0:MOVE 2*h
430     d i-1:TURNTO 45:MOVE h
440     a i-1
450   END IF
460 END DEFine a
470 :
480 DEFine PROCedure b(i)
490   IF i>0 THEN
500     b i-1:TURNTO 225:MOVE h
510     c i-1:TURNTO 270:MOVE 2*h
520     a i-1:TURNTO 315:MOVE h
530     b i-1
540   END IF
550 END DEFine b
560 :
570 DEFine PROCedure c(i)
580   IF i>0 THEN
590     c i-1:TURNTO 135:MOVE h
600     d i-1:TURNTO 180:MOVE 2*h
610     b i-1:TURNTO 225:MOVE h
620     c i-1
630   END IF
640 END DEFine c
650 :
660 DEFine PROCedure d(i)
670   IF i>0 THEN
680     d i-1:TURNTO 45:MOVE h
690     a i-1:TURNTO 90:MOVE 2*h
700     c i-1:TURNTO 135:MOVE h
710     d i-1
720   END IF
730 END DEFine d
```

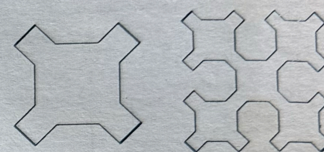


Fig.6-7: Sierpinski-Kurven S(1) u. S(2) (getrennt gezeichnet)

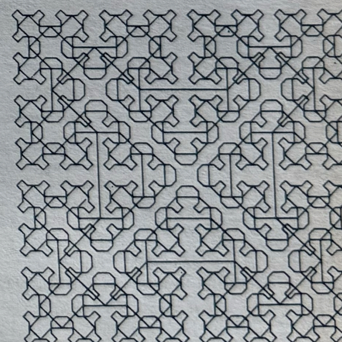


Fig.5: Sierpinski-Kurven S(1),...,S(4) (übereinander gezeichnet)

3. Fazit

Die beiden Programme zeigen recht deutlich, wie elegant mit Hilfe rekursiver Algorithmen unter Verwendung von Prozeduren in SuperBASIC des QL komplexe Kurvenmuster erzeugt werden können. Programmstruktur und -aufbau lassen dabei leicht sowohl die Richtigkeit als auch die Termination der Algorithmen erkennen. Letzteres auf Grund des Parameters i, der die Rekursion nicht beliebig groß werden läßt.

Eine Übertragung in andere BASIC-Versionen ohne rekursive Prozeduren führt zum Spaghetti-Code. Für PASCAL ergeben sich jedoch keine Schwierigkeiten.

Abschließend für den Interessierten eine interessante Aufgabe. Aus dem in Fig. 8 dargestellten Rekursionsmuster erstelle man ein Programm, das diese Kurven erzeugt.

Der Verfasser würde sich über Lösungen freuen. ■

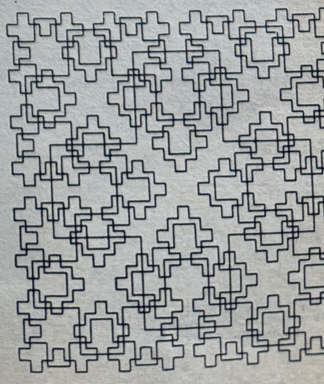


Fig.8: Wirth-Kurven W(1),...,W(4) (übereinander gezeichnet)

```
3750 PRINT #9,sdatei
3760 PRINT #9,sdatum
3770 PRINT #9,laenge
3780 FOR i=1 TO laenge
3790   PRINT#9,text(i)
3800 NEXT
3810 CLOSEOUT
3820 GOTO 3560
3830 IF s<>"1" THEN 3610
3840 CLS
3850 PRINT:PRINT "Druecken Sie bitte die PLAY-Taste am Recorder":PRINT
3860 CLS#1:PRINT#1,TAB(30);"Ich suche..."
3870 OPENIN "!supertext.file"
3880 INPUT #9,sfilename,sfiledatum,laenge
3890 PRINT "Gefunden: ";sfilename;", aufgezeichnet am ";sfiledatum;", Laenge:";
laenge;"Zeilen"
3900 PRINT#1," W)eitersuchen  L)aden  B)eenden"
3910 t=LOWER$(INKEY$):IF t="" THEN 3910
3920 IF t="w" THEN CLOSEIN:GOTO 3860
3930 IF t="b" THEN 4000
3940 IF t<>"l" THEN 3910
3950 CLS#1:PRINT#1,TAB(26);sfilename;" wird geladen"
3960 IF laenge>maxze THEN ERASE text:DIM text(laenge):maxze=laenge:PRINT:PRINT"
Die maximale Zeilenzahl wird auf"maxze" erhoeht !"
3970 FOR i=1 TO laenge
3980   LINE INPUT#9,text(i)
3990 NEXT
4000 CLOSEIN
4010 PRINT:PRINT"Druecken Sie die STOP-Taste am Recorder und eine beliebige Tas
te"
4020 CALL &BB06
4030 GOTO 3560
4040 '
4050 '        *** Beenden des Programms ***
4060 '
4070 INPUT "Sind Sie sicher, dass Sie das Programm beenden wollen (dadurch kann
Ihr Text    verloren gehen!) ? (Ja eingeben) ";s
4080 IF s<>"Ja" THEN RETURN
4090 IF szei<>"engl." THEN GOSUB 4300
4100 BORDER 0:PEN 1:PAPER 0
4110 CLS
4120 END
4130 '
4140 '    *** Deutscher Zeichensatz ***
4150 '
4160 PRINT#1,TAB(34);"Moment bitte"
4170 SYMBOL AFTER 90
4180 SYMBOL 91,&X1011010,&X111100,&X1100110,&X1100110,&X1111110,&X1100110,&X110
0110,&X0
4190 SYMBOL 92,&X10111010,&X1101100,&X11000110,&X11000110,&X11000110,&X1101100,
&X111000,&X0
4200 SYMBOL 93,&X1100110,&X0,&X1100110,&X1100110,&X1100110,&X1100110,&X111100,&
X0
4210 SYMBOL 123,&X1001000,&X0,&X1111000,&X1100,&X1111100,&X11001100,&X1110110,&
X0
4220 SYMBOL 124,&X100100,&X0,&X111100,&X1100110,&X1100110,&X1100110,&X111
0
4230 SYMBOL 125,&X1000100,&X0,&X1100110,&X1100110,&X1100110,&X1100110,&X1
&X0
4240 SYMBOL 126,&X111000,&X1101100,&X1101100,&X1101100,&X1100110,&X111011
01100,&X1100000
4250 KEY DEF 22,1,124,92
4260 KEY DEF 19,1,125,93
4270 KEY DEF 17,1,123,91
4280 KEY DEF 26,1,126,96
4290 RETURN
4300 PRINT#1,TAB(34);"Moment bitte"
4310 SYMBOL AFTER 240
4320 KEY DEF 26,1,&40,&7C
4330 KEY DEF 17,1,91,123
4340 KEY DEF 19,1,93,125
4350 KEY DEF 22,1,&5C,&60
4360 RETURN
4370 PRINT"                          Tastenbelegung mit deutschem Zeichensatz:"
4380 IF szei="deutsch" THEN GOSUB 4300
4390 PRINT:PRINT:PRINT
4400 LOCATE 35,5:PRINT"[ -"
4410 LOCATE 35,6:PRINT"\ -"
4420 LOCATE 35,7:PRINT"] -"
4430 LOCATE 35,8:PRINT"@ -"
4440 LOCATE 35,9:PRINT"{ -"
4450 LOCATE 35,10:PRINT"` -"
4460 LOCATE 35,11:PRINT"} -"
4470 GOSUB 4130:szei="deutsch"
4480 LOCATE 39,5:PRINT"{"
4490 LOCATE 39,6:PRINT"|"
4500 LOCATE 39,7:PRINT"}"
4510 LOCATE 39,8:PRINT"~"
4520 LOCATE 39,9:PRINT"["
4530 LOCATE 39,10:PRINT"\"
4540 LOCATE 39,11:PRINT"]"
4550 CLS#1
4560 PRINT:PRINT:PRINT:PRINT"E: einschalten"
4570 PRINT"A: ausschalten"
4580 PRINT:PRINT"  Bitte waehlen Sie!"
4590 s=UPPER$(INKEY$)
4600 IF s="A" THEN GOSUB 4300:szei="engl.":RETURN
4610 IF s="E" THEN RETURN
4620 GOTO 4590
```

Programmbeschreibung »Stadt in Not«

Von Jürgen Braun

Für Commodore 64

Der Planet Erde wird von UFO's eines feindlichen Planeten überfallen. Zur Abwehr der Angreifer steht nur noch ein letztes Lasergeschütz zur Verfügung. Dieses ist dem ständigen Angriff des Photonenhagels der feindlichen UFO's ausgesetzt.

Außerdem ist es abhängig von der Versorgung der einzelnen Munitionsdepots. Es ist daher äußerst wichtig, daß diese Depots solange als möglich unbeschädigt bleiben.

Eine Hilfe in diesem Kampf ist der Schutzschirm über den Versorgungslagern. Dieser bietet einen gewissen Schutz, aber nicht auf Dauer; wird durch ein Photonentorpedo ein Loch hineingesprengt, kann das darauffolgende Torpedo an dieser Stelle ungehindert hindurchfallen.

Wird dann ein Munitionsdepot von einem Torpedo getroffen, so geht dieses verloren und Sie können sich intensiv mit Ihrem Lasergeschütz um die Verteidigung der übriggebliebenen Depots kümmern.

Programmanleitung

Mit dem Ladeprogramm wird der Zeichensatz mittels Maschinenroutine umdefiniert (ca. 15 Sek.), das Hauptprogramm geladen und automatisch gestartet.

Danach erscheint das Titelbild. Die Zeit, die zum Einlesen der Sprites notwendig ist, wird durch die teilweise Auflösung des Titelbildes überbrückt.

Dann werden Sie aufgefordert, die Feuertaste zu drücken. In das Programm ist eine Joystickkennung eingebaut, so daß an beliebigem Controlport gespielt werden kann.

Sie können nun komfortabel mit dem Joystick zwischen „Start" und „Anleitung" wählen.

Jetzt beginnt Ihre Aufgabe, so lange wie möglich die vier Lager zu verteidigen, Ihr Geschütz darf dabei nicht getroffen werden, anderenfalls ist das Spiel sofort zu Ende.

Sie haben Ihre Mission verfehlt.

Den zur Verfügung stehenden Schutzschild können Sie reaktivieren, indem Sie den Joystick nach hinten ziehen.

Einen Bonusschutzschirm gibt es bei 20.000 und bei 70.000 Punkten. Wichtig: Zeitweise (zufällig) werden von den feindlichen UFO's Torpedos abgeschossen, die den Schutzschirm unbeschadet überwinden können. Es ist daher wichtig, möglichst alle Torpedos zu eliminieren.

Nach Ende des Spiels wird die erreichte Punktzahl angezeigt und es ertönt eine kleine Melodie, die zu einem neuen Spiel ermuntern soll. Sollten Sie mindestens 10.000 Punkte erreicht haben, können Sie sich in die Hi-Score-Liste eintragen. Erneuter „Start" oder „Ende" wird auch mit dem Joystick gewählt.

Anmerkung: Das Umschalten von zwei UFOsprites und deren Verbreiterung in X- und/oder Y-Richtung erzeugt den Eindruck, es mit acht verschiedenen UFO's zu tun zu haben.

Es wurden außerdem noch Warteschleifen eingefügt, die dazu beitragen, daß sich die Spielgeschwindigkeit bei mehr Bewegung nicht verringert. ■

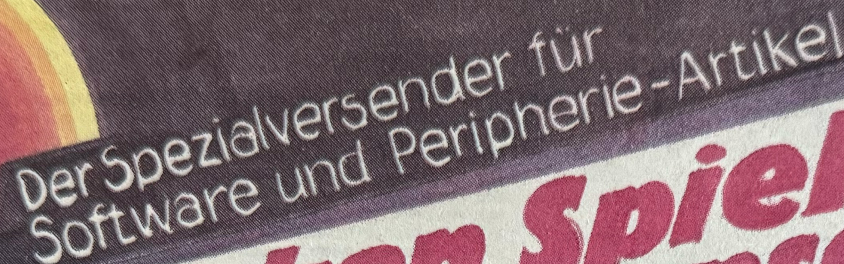
```
1 REM (C) + (E) BY PARASOFT 02.1985 / SPRITES BY A. HEYMANN
2 REM
3 REM DIE VERWENDETEN SONDERZEICHEN SEHEN WIE FOLGT AUS:
4 REM "[illegible]"  SHIFT/CLR  CRSR/DOWN  CTRL/GRN
5 REM UEBRIGENS: ES KOENNEN ALLE REM'S WEGGELASEN WERDEN
6 REM
7 POKE53281,0:POKE53280,0:PRINT"[illegible]LOAD TEIL II":GOSUB11
8 IFP1<>55507THENPRINT"FEHLER IN ZEILEN 19 - 63":STOP
9 PRINT"[illegible]LOAD"CHR$(34)"STADT 1.3"CHR$(34)",8":PRINT"[illegible]RUN"
10 POKE631,19:POKE632,13:POKE633,13:POKE634,13:POKE198,4:END
11 FORA=0TO34:READQ:P1=P1+Q:POKE828+A,Q:NEXT
12 IFP1<>6015THENPRINT"FEHLER IN ZEILE 12 - 15":STOP
13 DATA169,0,133,251,169,208,133,252,169,0,133,253,169,56,133,254,162,16,160
14 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251:SYS828:POKE1,PEEK(1)OR4
15 POKE56334,PEEK(56334)OR1:POKE53272,(PEEK(53272)AND240)OR14
16 P1=0:DATA0,177,251,145,253,200,208,249,230,252,230,254,202,208,240,96
17 READA:P1=P1+A:IFA=-1THENRETURN
18 POKE52,56:POKE56,56:FORJ=0TO7:READB:P1=P1+B:POKE14336+A*8+J,B:NEXT:GOTO17
19 DATA1,124,108,108,254,230,230,230,230
20 DATA2,252,108,108,126,118,118,118,254
21 DATA3,254,198,192,224,224,224,230,254
22 DATA4,254,102,102,118,118,118,118,254
23 DATA5,254,192,192,248,224,224,224,254
24 DATA6,254,192,192,248,224,224,224,224
25 DATA7,254,198,192,224,238,230,230,254
26 DATA0,60,66,153,161,161,153,66,60
27 DATA28,60,66,153,165,189,161,90,60
28 DATA43,0,24,24,126,126,28,28,0
29 DATA8,198,198,198,254,230,230,230,230
30 DATA9,48,48,48,56,56,56,56,56
31 DATA10,6,6,6,14,14,14,204,120
32 DATA11,198,204,216,254,230,230,230,230
33 DATA12,192,192,192,224,224,224,224,254
34 DATA13,198,238,254,214,198,230,230,230
35 DATA14,230,246,222,206,230,230,230,230
36 DATA15,254,198,198,230,230,230,230,254
37 DATA16,254,198,198,254,224,224,224,224
38 DATA17,254,198,198,230,230,236,246,0
39 DATA18,254,198,198,254,236,230,230,230
40 DATA19,254,198,192,254,14,14,206,254
41 DATA20,252,48,48,48,56,56,56,56
42 DATA21,198,198,198,230,230,230,230,254
43 DATA22,198,198,198,198,198,108,108,56
44 DATA23,230,230,230,198,214,254,238,198
45 DATA24,198,108,56,56,252,230,230,230
46 DATA25,198,198,198,198,124,56,56,56
47 DATA26,254,6,6,24,96,224,224,254
48 DATA48,254,198,206,222,246,230,230,254
49 DATA49,120,24,24,56,56,56,56,124
50 DATA50,254,198,6,254,224,224,224,254
51 DATA51,126,6,6,62,14,14,14,254
52 DATA52,192,192,204,204,254,28,28,28
53 DATA53,126,96,96,126,14,14,206,254
54 DATA54,254,192,192,254,230,230,230,254
55 DATA55,254,198,12,24,56,56,56,56
56 DATA56,124,108,108,254,230,230,230,254
57 DATA57,254,198,198,254,14,14,14,254
58 DATA33,48,48,48,56,56,56,0,56
59 DATA63,254,198,6,126,112,112,0,112
60 DATA44,0,0,0,0,120,120,24,24
61 DATA46,0,0,0,0,0,56,56,0
62 DATA61,0,0,126,0,126,126,0,0
63 DATA-1
READY.
```

```
1 REM (C) + (E) BY PARASOFT 02.1985 / SPRITES BY A. HEYMANN
2 REM DIE VERWENDETEN SONDERZEICHEN SEHEN WIE FOLGT AUS: (C= COMMODORE TASTE)
3 REM "█ █ █ █ █" C=/BLU CTRL/RVSON CTRL/RVSOFF CTRL/GRN SHIFT/CLR
4 REM "█ █ █ █ █" CTRL/H CRSR/DOWN CRSR/RIGHT CTRL/RED CLR
5 REM "█ █ █" CTRL/WHT CRSR/↑ CRSR/←
6 REM UEBRIGENS: ES KOENNEN ALLE REM'S WEGGELASSEN WERDEN
7 FORA=1TO5:S(A)=10000:B$(A)="@ BY PARASOFT":NEXT
8 POKE783,0:SS=1:V=53248:V1=V+21:B=1024:F=54272:BA=4:POKE53272,21
9 A$="..█/────────────────────────█..":B$="█ █":REM 34 * C=/Y
10 POKE53281,0:POKE53280,0:POKEV1,0:FORA=0TO24:POKEF+A,0:NEXT
11 PRINT"█//////  ////// ////// /////  //////":REM 2 1 1 2 SPACES
12 PRINT"█//       //   //  // //  //   //":REM 7 3 2 1 2 3 SPACES
13 PRINT"█ //      //   //  // //  //   //":REM 1 7 3 2 1 2 3 SPC
14 PRINT"█  //     //   ////// //  //   //":REM 2 6 3 1 2 3 SPC
15 PRINT"█   ///   //   //  // //  //   //":REM 3 4 3 2 1 2 3 SPC
16 PRINT"█//////   //   //  // /////    //█":POKE53272,PEEK(53272)OR14
17 REM ↑ 4 3 2 1 4 SPC
18 PRINTSPC(15)"█\\ \\\ \\"SPC(31)"\\ \\\ \\"SPC(31)"\\ \\\\\\"SPC(31)"\\ \\ \\\
19 PRINTSPC(15)"\\ \\ \\\"
20 PRINTSPC(9)"█/// // ////// //////"SPC(20)"/// // //  //   //":REM 1 1 2 3
21 PRINTSPC(9)"////// //  //   //"SPC(22)"// /// //  //   //":REM 1 2 3:1 1 2 3
22 PRINTSPC(9)"// /// //////   //█":REM 1 1 3
23 PRINT"█       @ + £  ":PRINT"█    B Y":PRINT"█PARASOFT 08.04"
24 REM ↑ 10 * "█" : 4 * SPC : 2 * SPC : 5 * SPC
25 PRINT"█":FORA=0TO35:READQ:NEXT
26 FORA=704TO704+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
27 FORA=832TO832+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
28 FORA=896TO896+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
29 FORA=960TO960+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
30 FORA=192*64TO192*64+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
31 FORA=193*64TO193*64+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
32 FORA=194*64TO194*64+62:READQ:POKEA,Q:POKEB+RND(1)*1000,32:NEXT
33 FORA=0TO3:POKE2040+A,11:NEXT
34 POKE2044,13:POKE2045,14:POKE2047,193:POKEV+28,223
35 FORA=0TO7:POKEV+39+A,5:NEXT:POKEV+44,7
36 POKEV+37,6:POKEV+38,2:POKE53280,6
37 PRINT"█"TAB(7)"█DRUECKE FEUER ZUM START"
38 J=PEEK(56321):IFJ=239THENJL=251:JR=247:FE=239:SU=253:PE=56321:RU=255:GOTO42
39 J=PEEK(56320):IFJ=111THENJL=123:JR=119:FE=111:SU=125:PE=56320:RU=127:GOTO42
40 GOTO38
41 FORA=1TO100:NEXT:J=PEEK(PE):IFJ=JRORJ<>JLGOTO43
42 PRINTTAB(7)"█ANLEITUNG       █START█  ":AN=1:GOTO41:REM 7 * SPC
43 IFJ=JRTHENPRINTTAB(7)"█ANLEITUNG       █START█  ":AN=0:REM 7 * SPC
44 IFJ=FEGOTO46
45 GOTO41
46 IFAN<>1GOTO62
47 PRINT"█SINN DES SPIELES BESTEHT DARIN, DIE VIER"
48 PRINT"BASEN VOR FEINDBOMBEN ZU SCHUETZEN. HIER"
49 PRINT"HILFT  EIN  SCHUTZSCHIRM, WELCHER  UEBER"
50 PRINT"DIE STADT GESPANNT IST.DOCH DURCH BOMBEN"
51 PRINT"KOENNEN LOECHER IN DEN SCHUTZSSCHIRM GE-"
52 PRINT"RISSEN WERDEN, WELCHE DIE  SCHUTZWIRKUNG"
53 PRINT"DES SCHIRMES BEEINTRAECHTIGEN.  <TASTE> ":POKEPE,0:POKEPE,0:WAITPE,16,1
6
54 PRINT"█SCHIESSEN SIE  DIE  UFOS  AB, BEVOR  SIE"
55 PRINT"IHRE TOEDLICHEN BOMBEN  ABWERFEN. VERSU-"
56 PRINT"CHEN SIE DIE STAEDTE  ZU  RETTEN UND ER-"
57 PRINT"SPIELEN  SIE  SICH  BONUSSCHUTZSCHIRME. "
58 PRINT"EIN DEFLEKTORSCHIRM  GIBT  ES  BEI 20000"
59 PRINT"UND BEI 70000 PUNKTEN. ICH WUENSCHE VIEL"
60 PRINT"SPASS UND VIEL GLUECK!!"
61 PRINT"█SCHUTZSCHILD REAKTIVIEREN = JOYST. DOWN":POKEPE,0:POKEPE,0:WAITPE,16,1
6
62 PRINT"█":GOSUB89:PRINT"█"B$SPC(36)B$B$SPC(36)B$B$SPC(36)"█ █"
63 POKE2023,160:POKEF+2023,14:GOSUB90:FORA=1TO7STEP2:POKEV+A,224:NEXT
64 POKEV1,15:POKEV+23,63:POKEV,60:POKEV+2,120:POKEV+4,200:POKEV+16,0:X4=180
65 POKEV+6,5:POKEV+29,15:POKEV+8,180:POKEV+9,194:POKEV1,PEEK(V1)OR16
66 POKEV+23,PEEK(V+23)OR128:U%=RND(1)*2:IFU%=0THENPOKE2046,15:GOTO68
67 POKE2046,192:POKEV+29,PEEK(V+29)AND191:POKEV+23,PEEK(V+23)AND191
68 U%=RND(1)*4+1:ONU%GOTO69,70,71,72
69 POKEV+29,PEEK(V+29)OR64:GOTO72
70 POKEV+23,PEEK(V+23)OR64:GOTO72
71 POKEV+29,PEEK(V+29)OR64:POKEV+23,PEEK(V+23)OR64
72 POKEV1,PEEK(V1)OR64:POKEV+16,PEEK(V+16)AND191
73 IFB0%=0THENB0%=1:B1%=INT(RND(1)*245+50):GOTO75
74 GOTO77
75 IFB1%>255THENPOKEV+16,PEEK(V+16)OR128:POKEV+14,B1%-255:GOTO77
76 POKEV+16,PEEK(V+16)AND127:POKEV+14,B1%
77 POKEV+13,RND(1)*100+60:FORU=0TO350STEPRND(1)*7+14:GOSUB98
78 J=PEEK(PE):IFJ=RUTHENFORA=1TO65:NEXT
79 IFB2%=0THENFORA=1TO30:NEXT
80 IFR=0THENFORA=1TO30:NEXT
81 IFJ=JLTHENX4=X4-13:GOSUB93
82 IFJ=JRTHENX4=X4+13:GOSUB93
83 IFJ=SUANDSS>0THENGOSUB89:SS=SS-1:GOSUB90
84 IFB2%=1THENGOSUB114
85 IFB0%=1ANDU>B1%ANDB2%=0THENB2%=1:HB=PEEK(V+13):POKEV+15,HB:POKEV1,PEEK(V1)OR1
28
86 IFR=1THENGOSUB100
87 IFR=0THENIFJ=FETHENPOKEV1,PEEK(V1)OR32:POKEV+16,PEEK(V+16)AND223:GOSUB100
88 NEXTU:POKEV1,PEEK(V1)AND191:GOTO66
89 POKE781,21:POKE782,0:SYS65520:PRINTA$:RETURN
90 PRINT"█PUNKTE  "P"█ SCHUTZSCHILDE  "SS:IFP>20000ANDS1=0THENS1=1:SS=SS+1
91 IFP>70000ANDS1=1THENS1=2:SS=SS+1
92 RETURN
93 IFX4<50THENX4=50
94 IFX4>252THENPOKEV+16,PEEK(V+16)OR16:GOTO96
95 POKE(V+16),PEEK(V+16)AND239:POKEV+8,X4:RETURN
96 IFX4>295THENX4=295
97 POKEV+8,X4-255:RETURN
98 IFU>255THENPOKEV+16,PEEK(V+16)OR64:POKEV+12,U-255:RETURN
99 POKEV+12,U:RETURN
100 IFX4>255ANDR=0THENRA=X4:POKEV+16,PEEK(V+16)OR32:POKEV+10,RA-255:RR=194:GOTO1
02
101 IFR=0THENRA=X4:POKEV+10,RA:RR=194
102 R=1:RR=RR-11:IFRR>0THENPOKEV+11,RR:GOTO104
103 R=0:RETURN
104 KO=PEEK(V+30):IFKO=96THENPOKEV1,PEEK(V1)AND159:R=0:GOTO107
105 IFKO=160THENPOKEV+30,0:GOTO163
106 RETURN
107 POKE2045,194:POKEV+11,PEEK(V+13)-15:POKEV1,PEEK(V1)OR32
108 IFU>255THENPOKEV+16,PEEK(V+16)OR32:POKEV+10,U-255:GOTO110
109 POKEV+10,U
110 POKEV+29,PEEK(V+29)OR32:W=44:GOSUB167:W=0
111 POKEV+44,5:POKEV+29,PEEK(V+29)AND223:POKEV+30,0:POKEV+11,255:POKEV+13,255
112 POKEV1,PEEK(V1)AND223:POKE2045,14:POKEV+12,0:POKEV+16,PEEK(V+16)AND223
113 P=P+INT(RND(1)*1000-2*U+500):POKEV+10,255:GOSUB90:U=345:RETURN
114 KO=PEEK(V+30):IFKO=129ORKO=130ORKO=132ORKO=136THENPOKEV+30,0:GOTO119
115 K1=PEEK(V+31):IFK1=128ANDHB>190THENPOKEV+31,0:GOTO160
116 IFKO=144THENPOKEV+30,0:GOTO131
117 HB=HB+9:IFHB>230THENPOKEV1,PEEK(V1)AND127:HB=0:B0%=0:B1%=0:B2%=0:RETURN
118 POKEV+15,HB:RETURN
119 POKEV1,PEEK(V1)AND127
120 IFKO=129THENPOKE2040,194:W=39:GOSUB167:POKEV1,PEEK(V1)AND254
121 IFKO=130THENPOKE2041,194:W=40:GOSUB167:POKEV1,PEEK(V1)AND253
122 IFKO=132THENPOKE2042,194:W=41:GOSUB167:POKEV1,PEEK(V1)AND251
123 IFKO=136THENPOKE2043,194:W=42:GOSUB167:POKEV1,PEEK(V1)AND247
124 POKEV+14,0:POKEV+15,0:HB=0:B0%=0:B1%=0:B2%=0:POKEV+46,5
125 BA=BA-1:IFBA>0THENRETURN
126 FORA=1TO4000:NEXT:POKEV+29,PEEK(V+29)OR16:POKE2044,194:W=43:GOSUB167
127 POKEV1,0:PRINT"█DA ALLE STAEDTE  ZERSTOERT  WURDEN, KON-"
128 PRINT"NTEN SIE KEINE  NEUE  MUNITION  ERWARTEN"
129 PRINT"UND  MUSSTEN  DEN  ROTEN  SELBSTZERSTOE-"
130 PRINT"RUNGSKNOPF BETAETIGEN.█":GOTO140
131 POKEV1,PEEK(V1)AND15:POKE2047,194:POKEV1,PEEK(V1)OR128
132 POKEV+29,PEEK(V+29)OR128:W=46:GOSUB167
133 POKEV1,PEEK(V1)AND127:W=39
134 IF(PEEK(V1)AND1)=1THENPOKE2040,194:GOSUB167:POKEV1,PEEK(V1)AND254
135 W=40:IF(PEEK(V1)AND2)=2THENPOKE2041,194:GOSUB167:POKEV1,PEEK(V1)AND253
136 W=41:IF(PEEK(V1)AND4)=4THENPOKE2042,194:GOSUB167:POKEV1,PEEK(V1)AND251
137 W=42:IF(PEEK(V1)AND8)=8THENPOKE2043,194:GOSUB167:POKEV1,PEEK(V1)AND247
138 POKEV1,0:PRINT"█LEIDER SIND DURCH DEN  VERLUST  DES  GE-"
139 PRINT"SCHUETZES ALLE STAEDTE VERNICHTET WORDEN"
140 PRINT"SIE HABEN ABER TROTZDEM "P" PUNKTE":PRINT"█ERREICHT.":GOSUB174
141 FORA=1TO1000:POKEPE,0:NEXT:POKE198,0
142 GOSUB148:PRINT"█NOCH EIN SPIEL?":PRINT"█"TAB(10)"█JA"SPC(10)"█NEIN":AN=1
143 J=PEEK(PE):IFJ=JRTHENPRINTSPC(10)"█JA"SPC(10)"█NEIN":AN=0
144 IFJ=JLTHENPRINTTAB(10)"█JA"SPC(10)"█NEIN":AN=1
145 IFJ=FEANDAN=1THENSS=1:BA=4:R=0:HB=0:B0%=0:B1%=0:B2%=0:P=0:GOTO33
146 IFJ=FETHENPRINT"█":POKE198,0:POKE54296,0:END
147 GOTO143
148 FORA=1TO5:IFP>S(A)THENFORA1=4TOASTEP-1:S(A1+1)=S(A1):B$(A1+1)=B$(A1):NEXT
149 IFP>S(A)THENPRINT"█SIE SIND UNTER DEN ERSTEN 5 !":INPUT"█IHR NAME BITTE";B1$
150 IFP>S(A)THENB1$=LEFT$(B1$+"             ",13):S(A)=P:B$(A)=B1$:GOTO152
151 NEXTA
152 FORA=1TO5:S$(A)=STR$(S(A)):S$(A)=LEFT$(S$(A)+"      ",6):NEXT
153 PRINT"█****************************************";:REM 40 * "*"
154 PRINT"***     HI-SCORE   HERO'S   LIST     ******"SPC(34)"***";:REM 5335 SPC
155 PRINT"***  PLATZ   NAME            PUNKTE  ***";:PRINT"***"SPC(34)"***";
156 REM ↑ 2 3 12 SPC
157 FORA=1TO5:PRINT"***  "A"    "B$(A)"  "S$(A)"   ***";:PRINT"***"SPC(34)"***"
;
158 REM ↑ 2 5 2 3 SPC
159 NEXT:PRINT"****************************************":RETURN:REM 40 * "*"
160 POKEV+23,PEEK(V+23)AND127:POKE2047,194:W=46:GOSUB167:POKE2047,193
161 POKEV1,PEEK(V1)AND127:HB=0:B0%=0:B2%=0:POKEV+14,0:POKEV+15,0
162 POKE781,21:SYS65520:PRINTTAB(B1%/8-3)"   █":B1%=0:RETURN
163 POKEV1,PEEK(V1)AND95:HB=0:B0%=0:B1%=0:B2%=0:POKEV+23,PEEK(V+23)AND127
164 POKE2047,194:POKEV1,PEEK(V1)OR128:POKEV+10,255:POKEV+11,255
165 W=46:GOSUB167:POKEV1,PEEK(V1)AND127:POKEV+23,PEEK(V+23)OR128
166 POKE2047,193:R=0:P=P+50:GOSUB90:POKEV+14,0:POKEV+15,0:POKEV+46,5:RETURN
167 POKEV+28,255:POKEF+24,15:POKEF+1,1:POKEF+5,11:POKEF+6,0:FORI=1TO150:NEXT
168 POKEF+12,9:POKEF+13,0:POKEF+19,10:POKEF+18,129
169 TH=30:FORFA=1TO16:FORB1=1TO15:POKEV+W,FA:FORI=1TO40:NEXT
170 POKEF+15,5:POKEF+8,TH
171 POKEF+15,3:POKEF+8,TH+20:TH=TH-1
172 NEXTFA
173 POKEF+4,0:POKEF+11,0:POKEF+18,0:POKEV+46,5:POKEV+28,223:RETURN
174 POKEF+3,8:POKEF+5,41:POKEF+6,89:POKEF+14,117:POKEF+18,16:POKEF+24,143
175 RESTORE
176 READFR,DR:IFFR=0THENRETURN
177 POKEF+4,65:FORT=1TODR*2:FQ=FR+PEEK(F+27)/2:HF=INT(FQ/256):LF=FQAND255
178 POKEF,LF:POKEF+1,HF:NEXT
179 POKEF+4,64:GOTO176
180 DATA4817,2,5103,2,5407,2:REM LIED
181 DATA8533,4,5407,2,8583,4
182 DATA5407,4,8583,12,9634,2
183 DATA10207,2,10814,2,8583,2
184 DATA9634,4,10814,2,8583,2
185 DATA9634,4,8583,12,0,0
186 DATA0,2,0:REM BASIS
187 DATA0,2,0
188 DATA0,2,0
189 DATA0,5,64
190 DATA0,10,128
191 DATA40,50,176
192 DATA160,50,176
193 DATA152,250,180
194 DATA171,250,191
```

Darstellung zum Artikel

Darstellung zum Artikel

```
185 DATA171,245,127
186 DATA85,85,85
197 DATA32,130,0
198 DATA130,0,130
189 DATA0,0,0
200 DATA0,0,0
201 DATA0,0,0
202 DATA0,0,0
203 DATA0,0,0
204 DATA0,0,0
205 DATA0,0,0
206 DATA0,0,0
207 DATA0,4,0:REM GESCHUETZ + RAKETE
208 DATA192,4,3
209 DATA128,63,86
210 DATA149,127,6
211 DATA140,63,6
212 DATA144,63,2
213 DATA130,107,162
214 DATA130,59,162
215 DATA130,25,34
216 DATA130,0,34
217 DATA21,85,84
218 DATA192,195,3
219 DATA0,0,0
220 DATA0,0,0
221 DATA0,0,0
222 DATA0,0,0
223 DATA0,0,0
224 DATA0,0,0
225 DATA0,0,0
226 DATA0,0,0
227 DATA0,0,0
228 DATA0,14,0:REM RAKETE SOLO
229 DATA0,14,0
230 DATA0,14,0
231 DATA0,14,0
232 DATA0,14,0
233 DATA0,0,0
234 DATA0,0,0
235 DATA0,0,0
236 DATA0,0,0
237 DATA0,0,0
238 DATA0,0,0
239 DATA0,0,0
240 DATA0,0,0
241 DATA0,0,0
242 DATA0,0,0
243 DATA0,0,0
244 DATA0,0,0
245 DATA0,0,0
246 DATA0,0,0
247 DATA0,0,0
248 DATA0,0,0
249 DATA0,60,0:REM UFO 1
250 DATA1,85,64
251 DATA85,170,85
252 DATA5,85,80
253 DATA1,85,64
254 DATA2,170,128
255 DATA2,0,128
256 DATA0,0,32
257 DATA0,0,0
258 DATA0,0,0
259 DATA0,0,0
260 DATA0,0,0
261 DATA0,0,0
262 DATA0,0,0
263 DATA0,0,0
264 DATA0,0,0
265 DATA0,0,0
266 DATA0,0,0
267 DATA0,0,0
268 DATA0,0,0
269 DATA0,0,0
270 DATA0,40,0:REM UFO 2
271 DATA0,160,0
272 DATA10,170,128
273 DATA169,85,168
274 DATA42,170,160
275 DATA5,85,64
276 DATA4,40,64
277 DATA16,0,16
278 DATA0,0,0
279 DATA0,0,0
280 DATA0,0,0
281 DATA0,0,0
282 DATA0,0,0
283 DATA0,0,0
284 DATA0,0,0
285 DATA0,0,0
286 DATA0,0,0
287 DATA0,0,0
288 DATA0,0,0
289 DATA0,0,0
290 DATA0,0,0
291 DATA0,204,0:REM BOMBE
292 DATA5,153,64
293 DATA1,153,0
294 DATA1,153,0
295 DATA0,160,0
296 DATA0,160,0
297 DATA0,40,0
298 DATA0,40,0
299 DATA0,40,0
300 DATA0,0,0
301 DATA0,0,0
302 DATA0,0,0
303 DATA0,0,0
304 DATA0,0,0
305 DATA0,0,0
306 DATA0,0,0
307 DATA0,0,0
308 DATA0,0,0
309 DATA0,0,0
310 DATA0,0,0
311 DATA0,0,0
312 DATA0,0,0,0:REM EXPLOSION
313 DATA32,0,0
314 DATA130,128,0
315 DATA32,128,2
316 DATA128,32,34
317 DATA5,0,0
318 DATA76,160,33
319 DATA221,40,133
320 DATA1,64,128
321 DATA204,40,33
322 DATA3,96,4
323 DATA3,26,131
324 DATA12,66,37
325 DATA243,10,161
326 DATA40,88,4
327 DATA20,0,35
328 DATA69,40,40
329 DATA10,32,2
330 DATA32,160,0
331 DATA130,0,0
332 DATA128,0
333 REM DATAPRUEFROUTINE MIT 'RUN 334' STARTEN,
    DANN 333- LOESCHEN.
334 FORA=1TO477:READQ:P1=P1+Q:NEXT
335 IFP1<>152358THENPRINT"DATENFEHLER !":STOP
336 PRINT"ALLES OK."
READY.
```

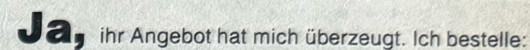


An die flinken Spieler in unserem Land

TOP-Schneider Software

House of Usher	29.—	Ghoostbusters	59.—
Manic Miner	29.—	Multidatei	69.—
Jet Set Willy	29.—	Multiadress	69.—
Flight Path 737	29.—	Multitext	89.—
American Football	49.—	Multivokabel	59.—
Fruity Frank	29.—	artwork	49.—
Survivor	29.—	Rocky Horror Show	39.—
Blagger (stereo)	35.—	Beach Head	49.—
Mission 1	39.—	Jump Jet	59.—
Chopper Squad	23.—	Nibbler	49.—
Dark Star	39.—	Money Molch	49.—
Hunchback	25.—	Mr. Pingo	49.—
Jack + Beanstalk	39.—	Time	49.—
Defend or die (st)	35.—	Minder	39.—
Flighter Pilot	29.—	Frankenstein	49.—
Moon Buggy	29.—	Super Pipeline 2	39.—
3D-Time Trek	29.—	Sorcery	45.—
Pyjamarama	25.—	Masterchess	35.—

SUPER BASICCOMPILER Cass. 79.— Disk 89.—



Ja, ihr Angebot hat mich überzeugt. Ich bestelle:

____ SUPER BASICCOMPILER Disk 89.—

____ SUPER BASICCOMPILER Cass. 79.—

Versandwünsche bitte angeben:
O Bargeld liegt bei O Verrechnungsscheck beigefügt
O per Nachnahme

Bei Versand per NN werden DM 5.— für Porto und Verpackung bei Aufträgen unter DM 100.— erhoben

NAME VORNAME

STRASSE PLZ/ORT

TELEFON UNTERSCHRIFT

Bitte auf Postkarte aufkleben und mit 60 Pf frankieren oder im Umschlag mit 80 Pf frankieren.

Bestellungen bitte an:

BILTEX — SOFTWARE
Kleine Schützenstraße 7, 5410 Höhr-Grenzhausen